

TEMA 4. Registro de la Información

1. Introducción. (2)
 - 1.1 Sistemas síncronos y asíncronos(3)
 - 1.2 Relojes. Características (4 - 10)
2. Almacenamiento estático de la información: Latches y Biestables (11 - 18)
 - 2.1 Entradas Asíncronas Preset y Clear (19)
 - 2.2 Latch tipo D (20 - 24)
 - 2.3 Biestables tipos D y T (25 - 47)
3. Especificaciones de los biestables. (48 - 52)
 - 3.1 Retardos. Tiempos de *set-up* y de *hold*. Frecuencia máxima f_{max}
 - 3.2 Hojas de características de la familia 74HCxx
4. Registros de desplazamiento uni y bidireccionales. (53-65)
5. Pilas *FIFO* y *LIFO*. (66 - 70)
6. Contadores. (71 - 77)

1

1. INTRODUCCIÓN

Los Sistemas Secuenciales son aquellos sistemas digitales en los que la salida en un instante dado, depende de las entradas en ese instante y de la **HISTORIA** del sistema, determinada por el **ESTADO**; es decir, serán sistemas con **MEMORIA**.

Para diseñar Sistemas Secuenciales es necesario utilizar elementos que memoricen el estado del sistema, que almacenen información.

Los sistemas secuenciales disponen de 4 tipos de señales: entrada, salida, variables de excitación y variables de estado; las dos últimas serán, las entradas y salidas de los elementos de memoria, respectivamente.

Los circuitos secuenciales están constituidos por circuitos combinacionales y elementos de memoria.

Sistemas de Almacenamiento de información:

- **CÉLULAS ESTÁTICAS**: están basadas en realimentación. **BIESTABLES**
- **CÉLULAS DINÁMICAS**: basadas en el almacenamiento de carga en la capacidad de puerta de los dispositivos MOS. **PUERTA DE TRANSMISIÓN -INVERSOR**

2

1.1 SISTEMAS SÍNCRONOS Y ASÍNCRONOS

Los sistemas secuenciales se pueden clasificar de acuerdo con la respuesta respecto a una señal de reloj en **Síncronos** y **Asíncronos**.

Síncronos son aquellos en los que existe al menos una señal de reloj que va marcando en que momento se pueden producir los cambios en el circuito: sincroniza el funcionamiento del circuito.

Son sistemas en los que los cambios en los estados se producen controlados por una señal de activación a través de una entrada especial del sistema, denominada "entrada de reloj"

Asíncronos son aquellos que responden autónomamente a los cambios de las entradas

3

1.2 RELOJES: CARACTERÍSTICAS (I)

Una señal de reloj se genera con lo que se conoce como OSCILADOR A CRISTAL.

Se construye con material piezoeléctrico → bajo una corriente alterna de cierta frecuencia, el cristal oscila y viceversa: si el cristal oscila produce una corriente alterna (AC).

Frecuencia = k/L

k = cte dependiente del corte del cristal y L = anchura

La frecuencia de estos osciladores varía con el tiempo debido a:

- **Envejecimiento**
- **Efectos de la temperatura**
- **Otras causas.**

(en los cristales utilizados para la fabricación de relojes y computadores se tiene una variación de 1 ppm al día ($1/10^6$ o sea 0,0001 %)).

Los materiales mas utilizados son:

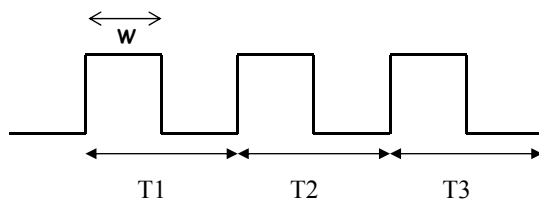
- **Turmalina (silicato de Na, Li, Mg, Fe...): mas resistente y cara.**
- **Cuarzo: mas abundante y barato. Funciona bien hasta 10 MHz. Para frecuencias mayores (del orden de 100MHz) hay que montar cristales de otros tipos.**
- **Sales de Rochelle (tartrato doble de Na y K) $\text{KNaC}_4\text{H}_4\text{O}_6 \cdot 4\text{H}_2\text{O}$.**

1.2 RELOJES: CARACTERÍSTICAS (II)

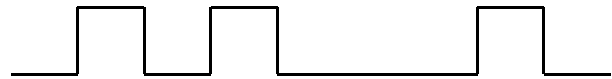
Las señales de reloj son un conjunto (tren) de pulsos normalmente simétricos; es decir, un conjunto de valores altos (1) y bajos (0), continuo en el tiempo.

Cuando los intervalos de tiempo son fijos entre ellos se forma un **tren periódico**, que queda definido mediante el valor de su periodo (T) o el de su inversa, la frecuencia (f)

Si no tiene repetición de pulsos en forma periódica, se obtiene un tren de pulsos **no periódico**.



Tren de pulsos periódico

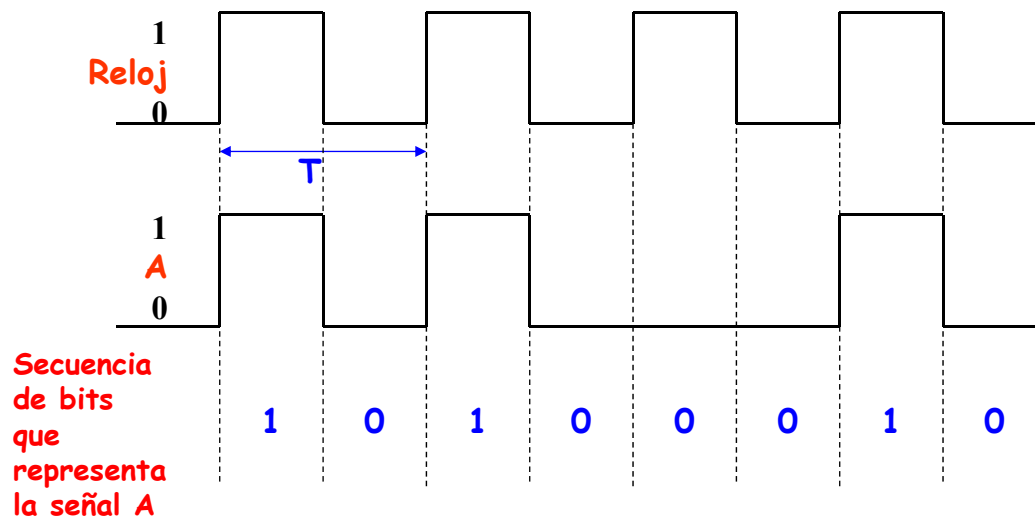


Tren de pulsos no periódico

5

1.2 RELOJES: CARACTERÍSTICAS (III)

CRONOGRAMA o DIAGRAMA DE TIEMPOS: es una ilustración o gráfica de ondas digitales que muestra la relación temporal entre varias señales.



Cronograma con señal de reloj y señal A

6

1.2 RELOJES: CARACTERÍSTICAS (IV)

CARACTERÍSTICAS DEL PULSO DE RELOJ :

- Anchura del pulso (W)
- Periodo (T)
- Frecuencia (F),
 $F = 1/T$

* ciclo de trabajo

$$\text{Ciclo de trabajo} = W/T$$

Atendiendo a la "forma de trabajo" de los circuitos con respecto al reloj:

- activación por flanco
- activación por nivel

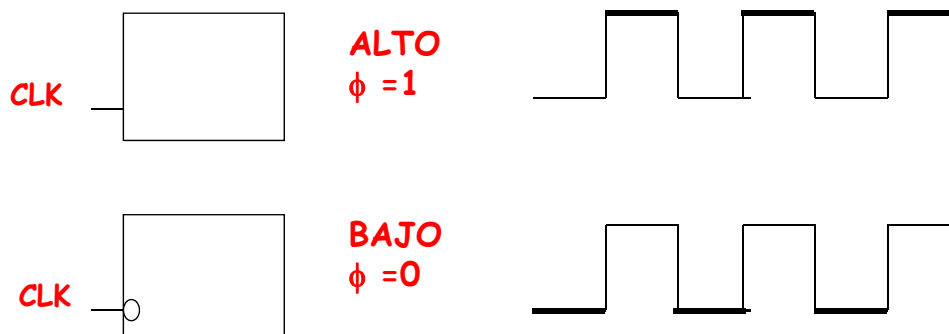
7

1.2 RELOJES: CARACTERÍSTICAS (V)

- ACTIVACIÓN o FUNCIONAMIENTO por: **NIVEL**
La señal de reloj debe alcanzar el nivel de tensión

Ventajas: La información se transmite durante un tiempo considerable.

Inconvenientes: Posibles fluctuaciones en los valores de la tensión ("guiños").

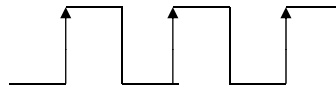
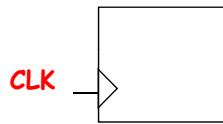


8

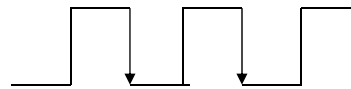
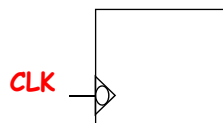
1.2 RELOJES: CARACTERÍSTICAS (VI)

- ACTIVACIÓN o FUNCIONAMIENTO por: **FLANCO**
Solamente actúa en las transiciones

POSITIVO O DE SUBIDA



NEGATIVO O DE BAJADA



Ventajas: Funciona en las transiciones y se evitan "guiños".

Inconvenientes: Que la pendiente de la señal sea elevada y no de tiempo suficiente para que pase la información.

9

1.1 RELOJES: CARACTERÍSTICAS (VII)

TEMPORIZACIÓN: señales de reloj

- ESTRATEGIAS DE TEMPORIZACIÓN:
 - Una fase de reloj Φ
 - Doble fase de reloj $\Phi \overline{\Phi}$
 - Pseudo doble fase o dos fases sin solapamiento Φ_1 y Φ_2
 - Cuatro fases $\Phi_1 \Phi_2 \overline{\Phi_1} \overline{\Phi_2}$
 - Pseudo cuatro fases $\Phi_1 \Phi_2 \Phi_3 \Phi_4$

10

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (I)

BIESTABLE: Unidad elemental de almacenamiento, que permite almacenar 1 bit; puede almacenar dos estados estables "0" y "1": La información almacenada puede ser recuperada (leída) y la lectura no es destructiva.

- **LATCHES (CERROJOS)**
 - Asíncronos
 - Implementación con inversores
 - RS NOR, RS NAND
 - Síncronos (por nivel)
 - RS NOR, RS NAND
 - JK, D y T
- Con entradas asíncronas: PRESET (**PR**) y CLEAR (**CLR**)
- **FLIP-FLOPS** (Síncronos por flanco)
 - Master-Slave (Amo-Esclavo)
 - D, SR, JK, T

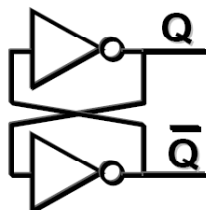
Unidad de varios bits: **REGISTRO**

11

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (II)

LATCH CON INVERSORES ASÍNCRONO (histórico)

La idea básica de la célula estática se puede mostrar en un circuito formado por dos inversores: realimentación



Si $Q = 0 \rightarrow \bar{Q} = 1$

Si $Q = 1 \rightarrow \bar{Q} = 0$

Realimentación: la señal de salida se realimenta a la entrada, conservando el valor

Dos **estados estables** Q y \bar{Q}

No podemos elegir en **cual de los estados posibles** se va a encontrar el biestable

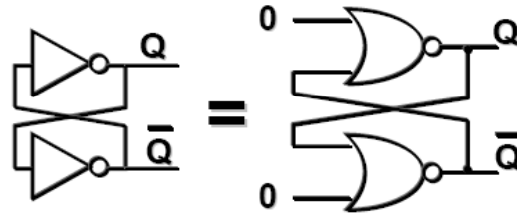
No podemos **modificar el estado**: hemos de añadir 2 señales externas

12

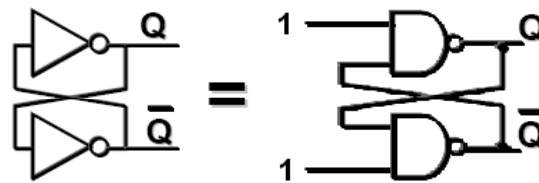
2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (III)

Para resolver las limitaciones de la implementación anterior, se plantea la utilización de otras puertas lógicas con mayor número de entradas

Puertas NOR



Puertas NAND

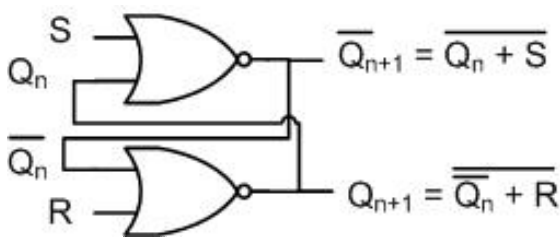


13

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (IV)

LATCH SR (set-reset) (activar-desactivar).

Construido con 2 puertas **NOR** con **ACOPLAMIENTO CRUZADO**



S R	Q_{n+1}	\overline{Q}_{n+1}
0 0	Q_n	\overline{Q}_n
0 1	0	1
1 0	1	0
1 1	0*	0*

S R	Q_n	\overline{Q}_n	Q_{n+1}	\overline{Q}_{n+1}	Q_{n+1}	\overline{Q}_{n+1}	
0 0	0	1	0	1	Q_n	\overline{Q}_n	MANTIENE
0 0	1	0	1	0			
0 1	0	1	0	1	0 (1 ciclo)		DESACTIVA:
0 1	1	0	0	0			RESET
0 1			0	1		1 (2 ciclos)	
1 0	0	1	0	0		0 (1 ciclo)	ACTIVA:
1 0			1	0			SET
1 0	1	0	1	0	1 (2 ciclos)		
1 1	0	1	0	0			INDETERMI-
1 1			0	0			NADO
1 1	1	0	0	0	0*	0*	NO USADO
1 1			0	0			

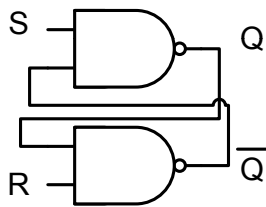
RESET es PONER a 0
SET es PONER a 1

Asíncronos: Las entradas actúan continuamente y las salidas están disponibles inmediatamente (tras retardos del circuito).

14

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (V)

LATCH SR. Construido con 2 puertas **NAND** con **ACOPLAMIENTO CRUZADO**



S	R	Q	Q _{next}	\overline{Q}_{next}
0	0	x	1	1
0	1	x	1	0
1	0	x	0	1
1	1	0	0	1
1	1	1	1	0

\overline{S}	\overline{R}	S	R	Q	Q _{next}	\overline{Q}_{next}	
1	1	0	0	x	1	1	Indeterminación
1	0	0	1	x	1	0	
0	1	1	0	x	0	1	
0	0	1	1	0	0	1	
0	0	1	1	1	1	0	

$$Q_{n+1} = S \cdot \overline{Q_n}$$

$$\overline{Q}_{n+1} = R \cdot Q_n$$

Si R=1 Q_{next} = 1
Si R=0 Q_{next} = 0

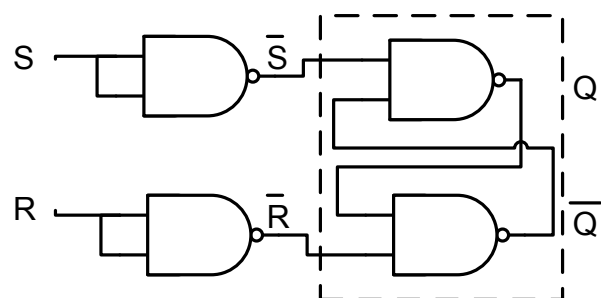
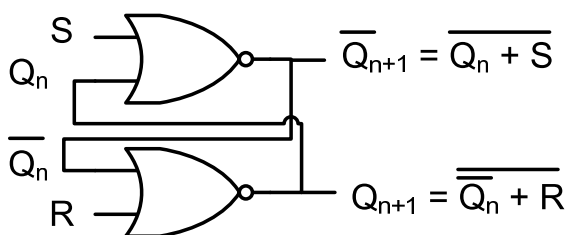
Como RESET es PONER a 0
SET es PONER a 1
La ACTIVACION POR las NEGADAS

Asíncronos. Las entradas actúan continuamente y las salidas están disponibles inmediatamente (tras retardos del circuito).

15

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (VI)

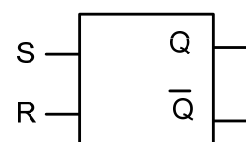
LATCH SR: RESUMEN



S	R	Q	Q _{next}	\overline{Q}_{next}
0	0	0	0	1
0	0	1	1	0
0	1	x	0	1
1	0	x	1	0
1	1	x	1	1

Tabla universal de Latch SR
Independiente del tipo de puertas
con las que se implementa

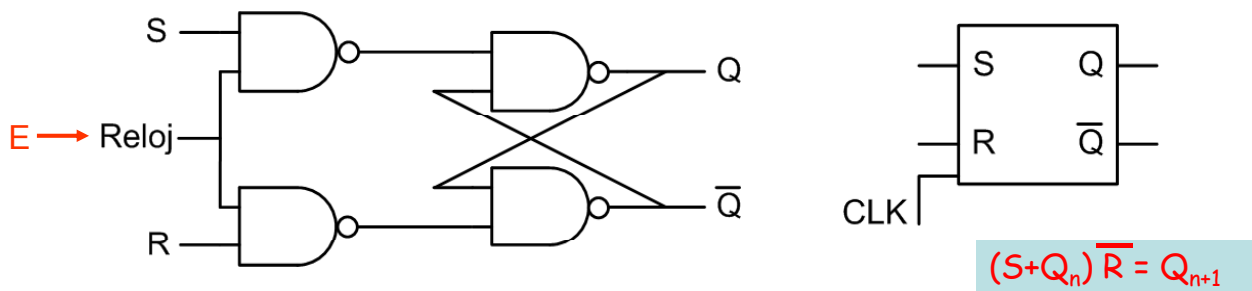
S	R	Q(t+1)	$\overline{Q}(t+1)$
0	0	Q(t)	$\overline{Q}(t)$
0	1	0	1
1	0	1	0
1	1	—	—



16

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (VII)

LATCH SR NAND "SÍNCRONO" POR NIVEL ALTO (I)



S/R	Reloj CLK	salida
0	0	1
0	1	1
1	0	1
1	1	*

Si Reloj = CLK = 0 → Salida de las puertas NAND es "1" se mantiene el estado

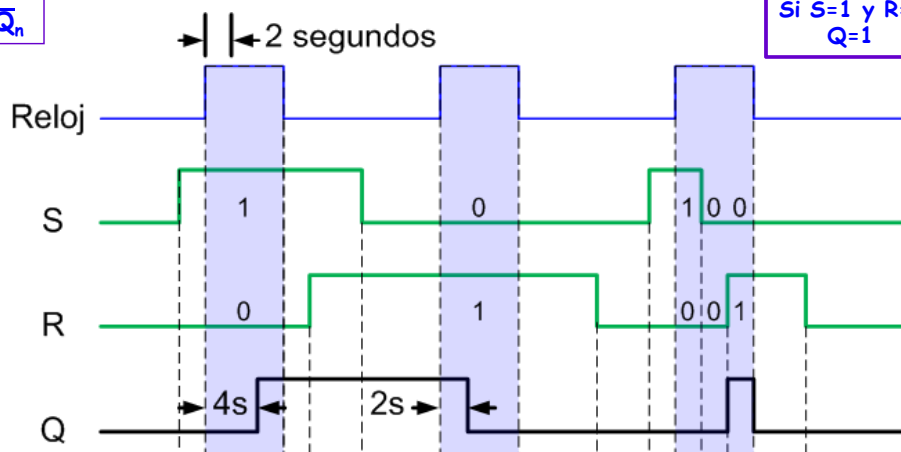
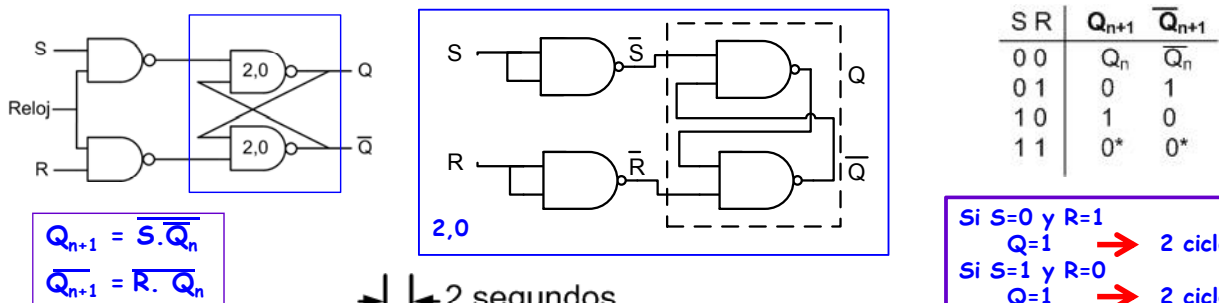
Si Reloj = CLK = 1 → Salida de las puertas NOR es \bar{S} y \bar{R} , respectivamente →

COMPORTAMIENTO IDÉNTICO AL MODELO ANTERIOR

17

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (VIII)

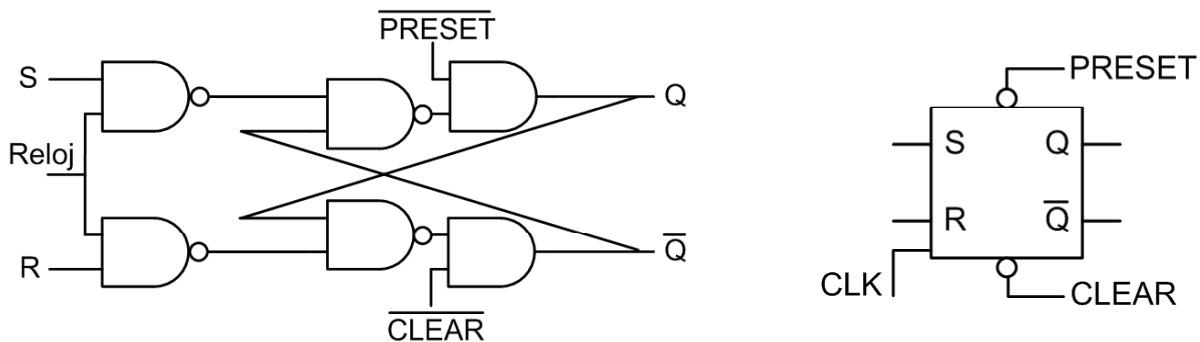
LATCH SR NAND "SÍNCRONO" POR NIVEL ALTO (II)



18

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (IX)

LATCH SR NAND SÍNCRONO POR NIVEL ALTO CON ENTRADAS ASÍNCRONAS PRESET Y CLEAR (III)



ENTRADAS ASÍNCRONAS: tienen prioridad sobre las entradas síncronas de los biestables

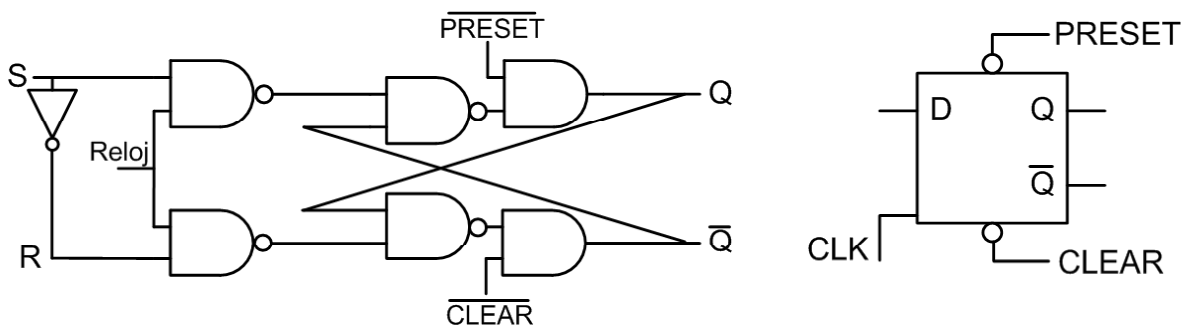
PRESET: puesta a uno (si $PR = 1$ $\underline{Q} = 0$)

CLEAR: puesta a cero (si $CL = 1$ $\underline{Q} = 0$)

19

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (X)

2.2 LATCH D NAND SÍNCRONO POR NIVEL ALTO CON ENTRADAS ASÍNCRONAS PRESET Y CLEAR (I)



S	R	D	Q_{n+1}	\bar{Q}_{n+1}
0	1	0	0	1
1	0	1	1	0

$CLR = 1$ $Q = 0$

$PRS = 1$ $Q = 0$

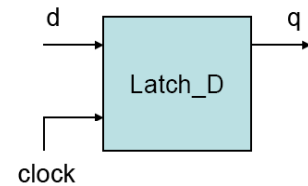
20

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XI)

2.2 LATCH D (síncrono por nivel) (II)

Con reloj:

- LATCH disparados por NIVEL



CLK	D	Q	Q _{next}	\overline{Q}_{next}	Modo
1	0	x	0	1	Puesta a 0
1	1	x	1	0	Puesta a 1
0	x	x	No cambia		Inactivo

MODELADO CON VHDL

--Entidad

```

LIBRARY IEEE;
USE IEEE.STD_LOGIC_1164.all;

ENTITY latch_D IS
    GENERIC (retardo:TIME:= 0 NS);
    PORT (clock,d: IN STD_LOGIC; q: OUT STD_LOGIC);
END latch_D;
    
```

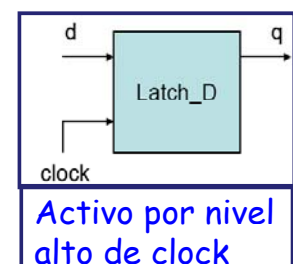
21

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XII)

2.2 LATCH D (síncrono por nivel) (III)

CLK	D	Q	Q _{next}	\overline{Q}_{next}	Modo
1	0	x	0	1	Puesta a 0
1	1	x	1	0	Puesta a 1
0	x	x	No cambia		Inactivo

MODELADO CON VHDL



-- Arquitectura en estilo de comportamiento

```

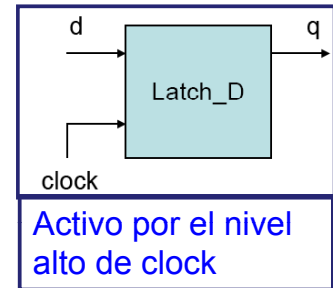
-----
ARCHITECTURE comporta_latch OF latch_D IS
BEGIN
    PROCESS (clock,d)
    BEGIN
        IF clock = '1' THEN -- activo a nivel alto
        -- IF clock = '0' THEN -- activo a nivel bajo
            q<= d AFTER retardo;
        END IF;
    END PROCESS;
END comporta_latch;
    
```

22

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XIII)

2.2 LATCH D (síncrono por nivel) (IV)

CLK	D	Q	Q _{next}	\overline{Q}_{next}	Modo
1	0	x	0	1	Puesta a 0
1	1	x	1	0	Puesta a 1
0	x	x	No cambia		Inactivo



MODELADO CON VHDL

-- Arquitectura en estilo de flujo de datos
-- Activo a nivel alto

ARCHITECTURE flujo OF latch_D IS
BEGIN

 q <= d AFTER retardo WHEN clock = '1'; -- activo a nivel alto
 --q <= d AFTER retardo WHEN clock = '0'; -- activo a nivel bajo
END flujo;

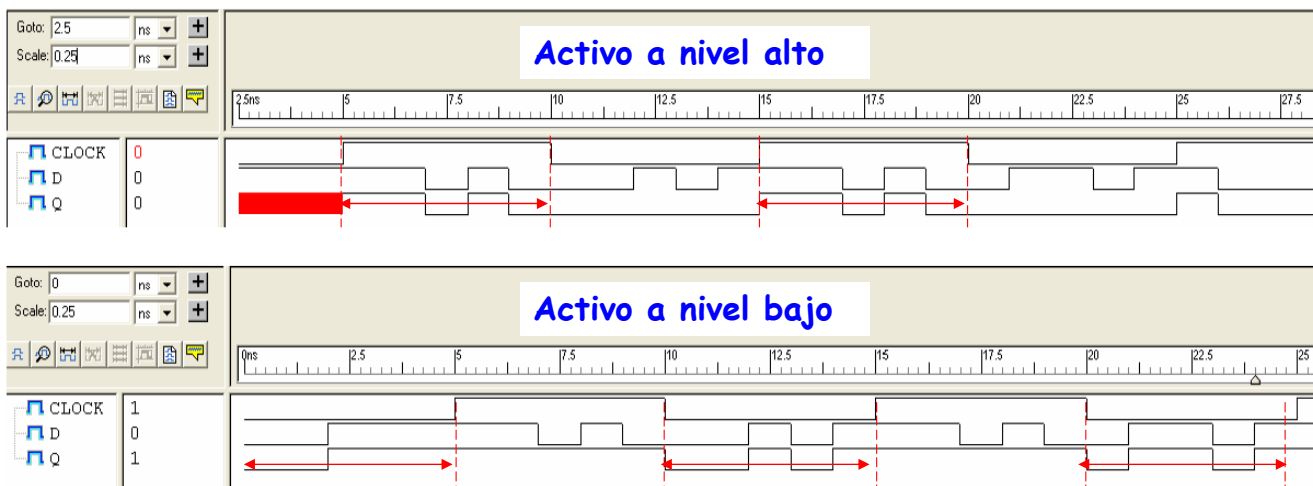
23

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XIV)

2.2 LATCH D (síncrono por nivel) (V)

CLK	D	Q	Q _{next}	\overline{Q}_{next}	Modo
1	0	x	0	1	Puesta a 0
1	1	x	1	0	Puesta a 1
0	x	x	No cambia		Inactivo

RESULTADOS DE SIMULACIÓN

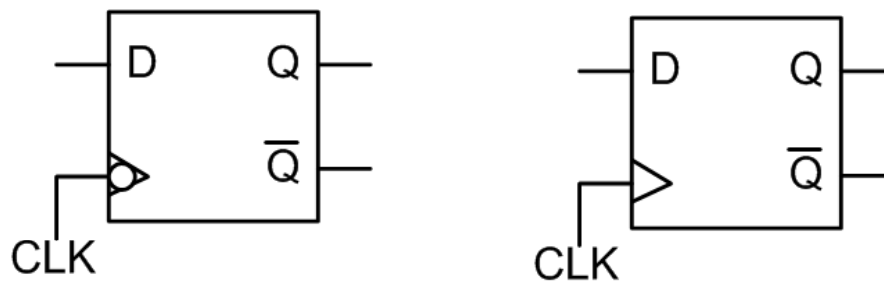


NOTA: Los valores que se presentan a la entrada se reflejan en la salida durante el tiempo que la señal de reloj está activa

24

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XV)

2.3 FLIP FLOP D (biestable síncrono por flanco) (I)



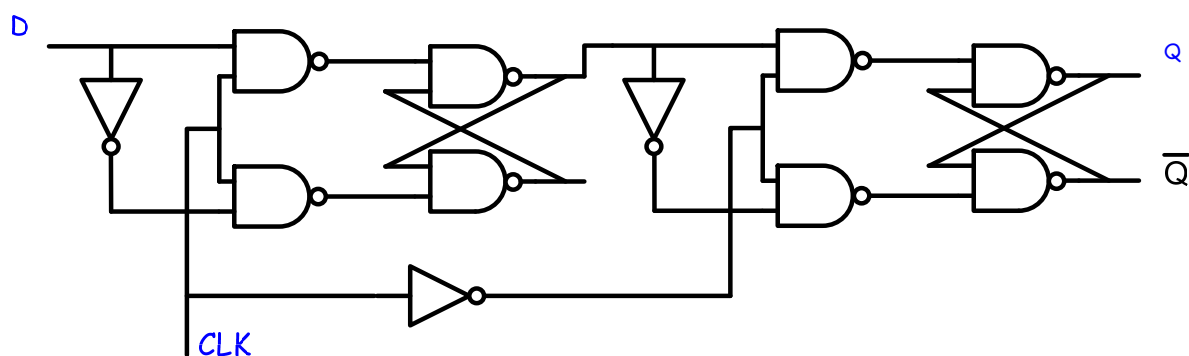
D	CLK	Q(+)	$\overline{Q}(+)$	Comentario
0	↓↑	0	1	RESET (almacena un 0)
1	↓↑	1	0	SET (almacena un 1)

25

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XVI)

2.3 FLIP FLOP D (biestable D síncrono por flanco MASTER SLAVE) (II)

DOS LATCHES SR CONCATENADOS



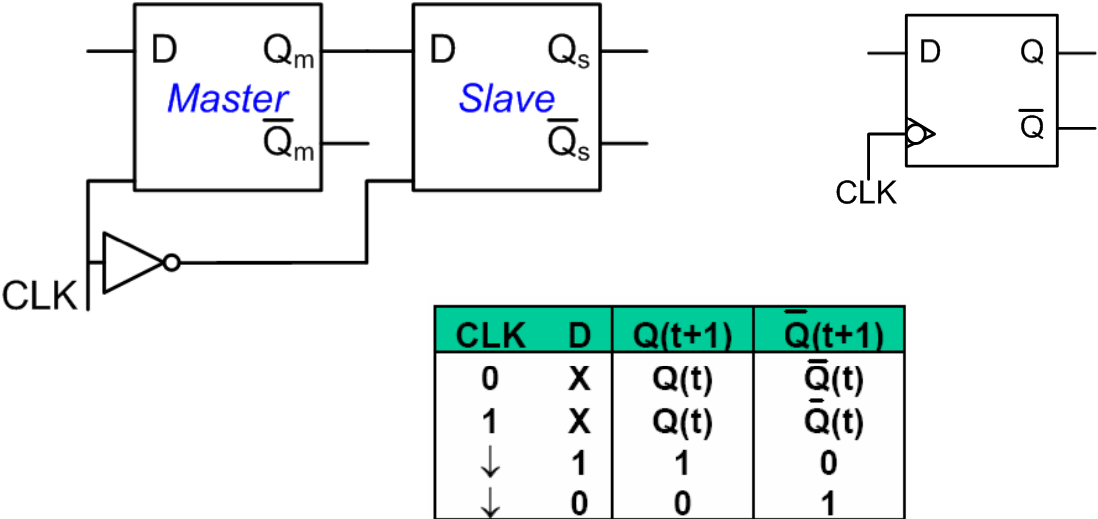
26

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XVII)

2.3 FLIP FLOP D (biestable D síncrono por flanco) (III)

Si Reloj = CLK= 0 mantiene estado en 1º D

FLANCO DE BAJADA ↴

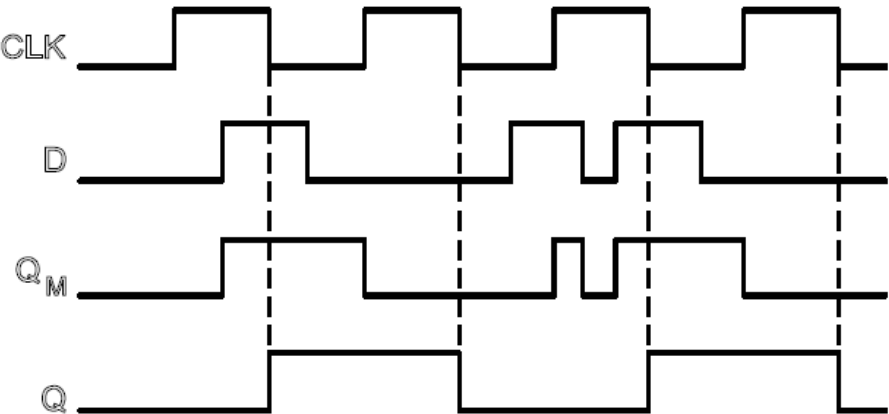


2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XVIII)

2.3 FLIP FLOP D (biestable D síncrono por flanco) (IV)

FLANCO DE BAJADA ↴

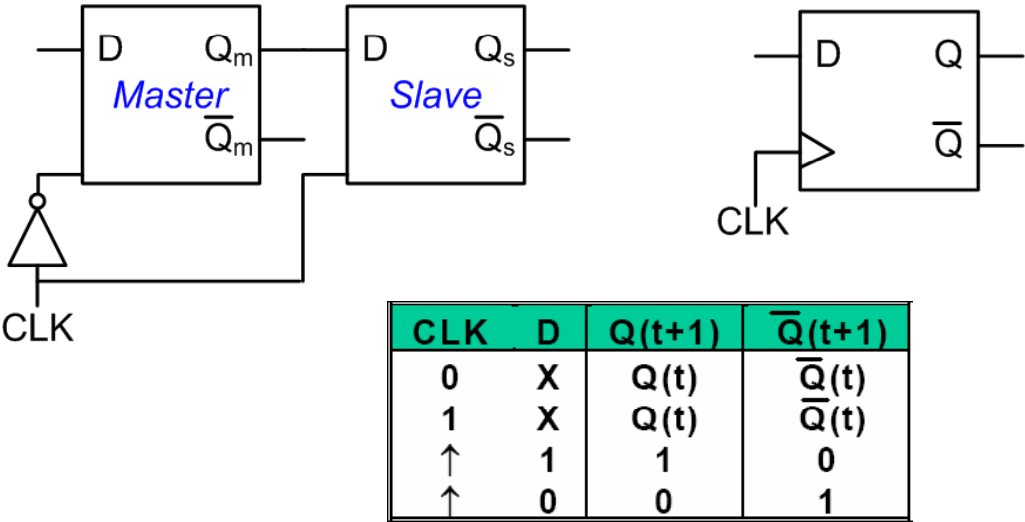
CLK	D	Q(t+1)	$\overline{Q}(t+1)$
0	X	Q(t)	$\overline{Q}(t)$
1	X	Q(t)	$\overline{Q}(t)$
↓	1	1	0
↓	0	0	1



2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XIX)

2.3 FLIP FLOP D (biestable D síncrono por flanco) (V)

FLANCO DE SUBIDA ↗

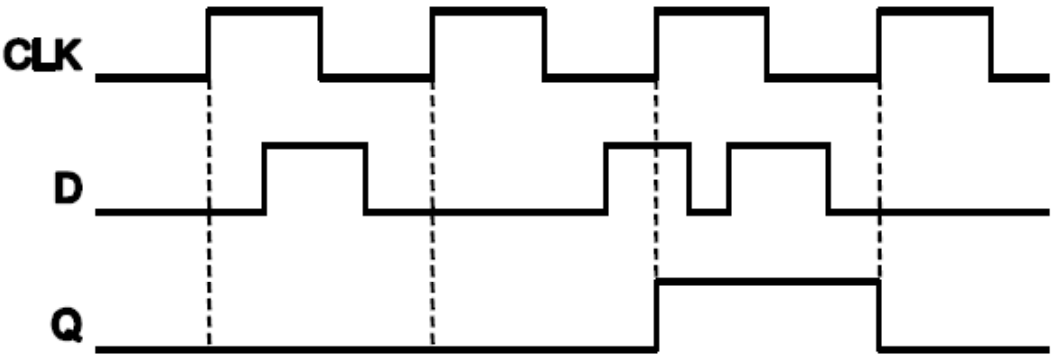


2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XX)

2.3 FLIP FLOP D (biestable D síncrono por flanco) (VI)

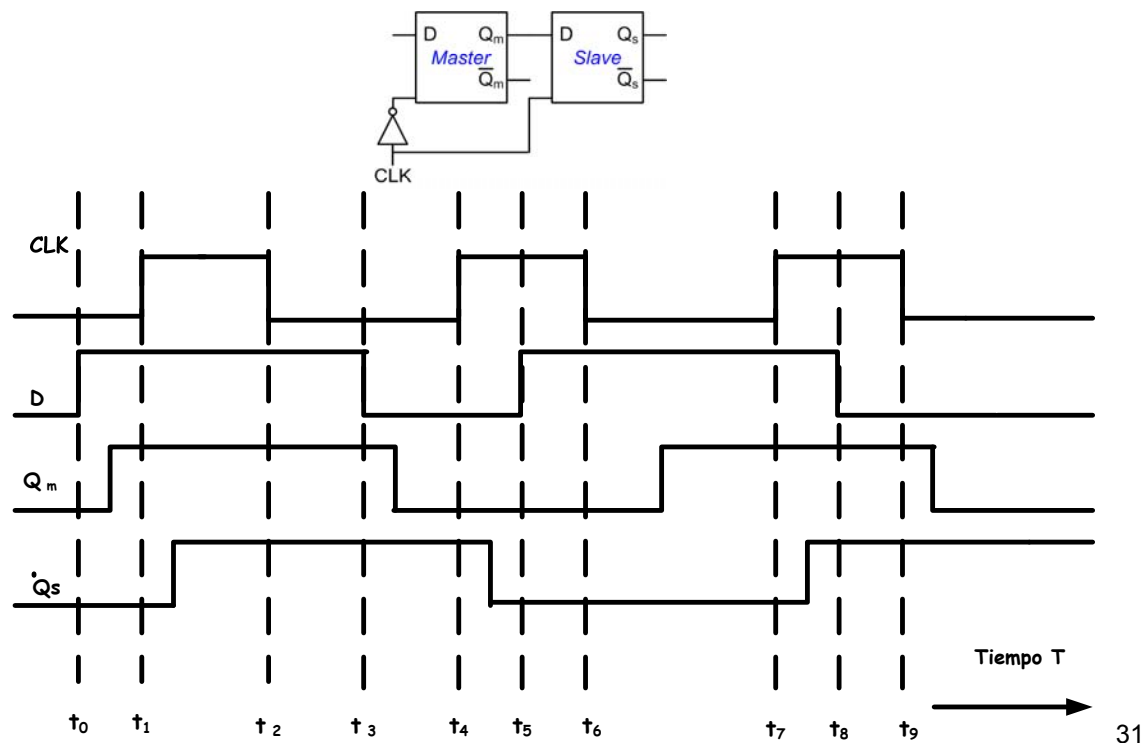
FLANCO DE SUBIDA ↗

CLK	D	$Q(t+1)$	$\overline{Q}(t+1)$
0	X	$Q(t)$	$\overline{Q}(t)$
1	X	$Q(t)$	$\overline{Q}(t)$
↑	1	1	0
↑	0	0	1



2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXI)

2.3 FLIP FLOP D (biestable D síncrono por flanco de subida) (VII)

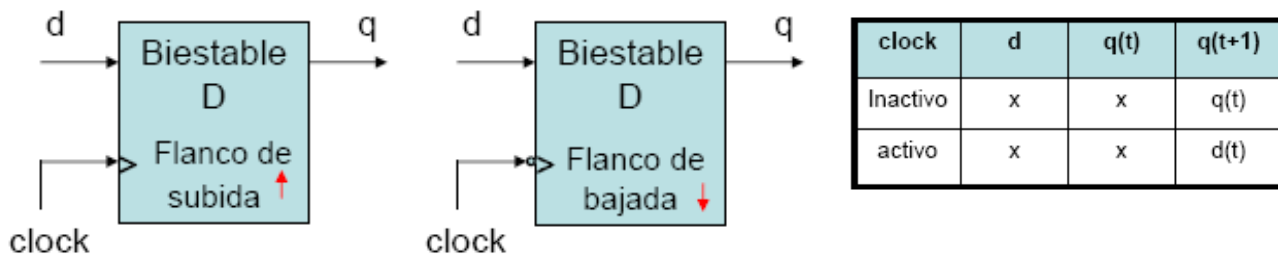


31

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXII)

2.3 FLIP FLOP D (VIII)

MODELADO CON VHDL



```
--Entidad
LIBRARY IEEE;
USE ieee.STD_LOGIC_1164.all;

ENTITY biestable_D IS
    GENERIC (retardo:TIME:=0 ns);
    PORT (clock, d: IN STD_LOGIC; q: OUT STD_LOGIC);
END biestable_D;
```

32

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXIII)

2.3 FLIP FLOP D (IX)

MODELADO CON VHDL

```
-- Arquitectura en estilo de COMPORTAMIENTO

-- Reloj Activo por flanco de subida [rising_edge (clock)]
-- Reloj Activo por flanco de bajada [falling_edge (clock)]

-----

ARCHITECTURE comporta_biestable OF biestable_D IS
BEGIN
  PROCESS (clock)
  BEGIN
    -- IF rising_edge (clock) THEN  --Flanco de subida
    IF falling_edge (clock) THEN  --Flanco de bajada
      q<=d AFTER retardo;
    END IF;
  END PROCESS;
END comporta_biestable;
```

33

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXIV)

2.3 FLIP FLOP D (X)

MODELADO CON VHDL

```
-- Arquitectura en estilo de FLUJO DE DATOS

-----

ARCHITECTURE flujo OF biestable_D IS
BEGIN

  q <= d AFTER retardo WHEN falling_edge (clock);  --Flanco de bajada

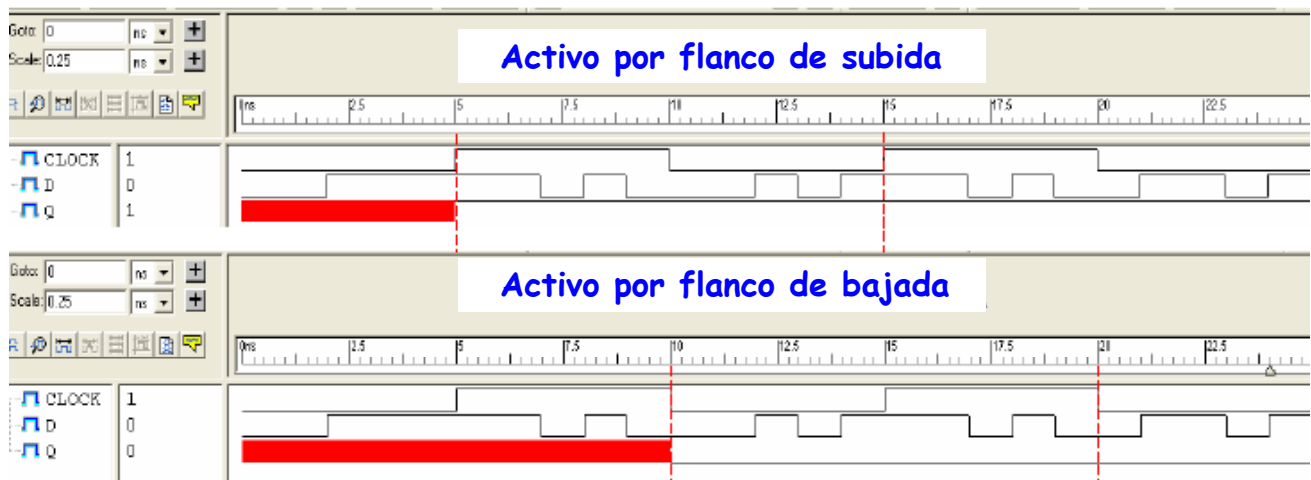
END flujo;
```

34

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXV)

2.3 FLIP FLOP D (XI)

RESULTADOS DE SIMULACIÓN



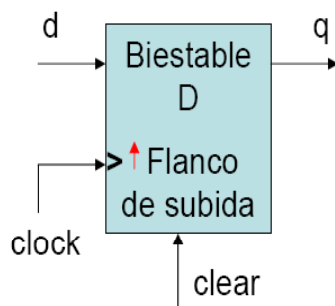
NOTA: los valores que se presentan a la entrada durante el flanco del reloj son los que se reflejan a la salida.

35

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXVI)

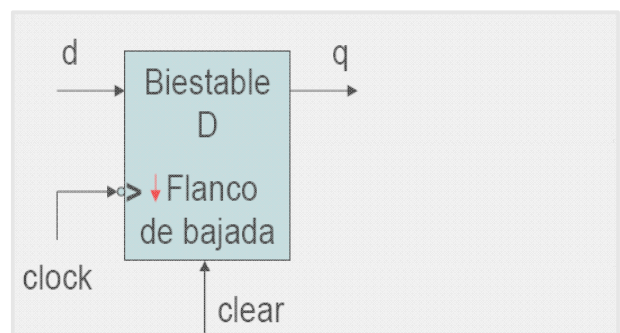
2.3 FLIP FLOP D con entradas asíncronas (CLEAR) (XII)

MODELADO CON VHDL



clear asincrono

clock	clear	d(t)	q(t)	q(t+1)
inactivo	inactivo	x	x	q(t)
activo	inactivo	x	x	d(t)
x	activo	x	x	'0'



clear sincrono

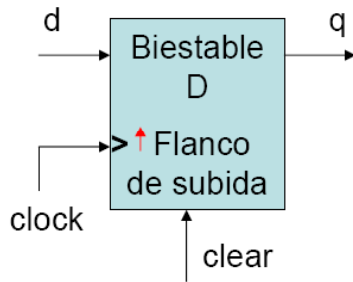
clock	clear	d(t)	q(t)	q(t+1)
inactivo	inactivo	x	x	q(t)
activo	inactivo	x	x	d(t)
activo	activo	x	x	'0'

36

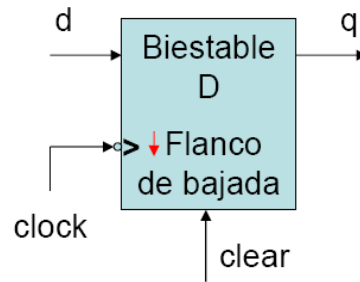
2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXVII)

2.3 FLIP FLOP D con entradas asíncronas (CLEAR) (XIII)

MODELADO CON VHDL



rising_edge (clock)



falling_edge (clock)

--Entidad

```
LIBRARY IEEE;
USE ieee.STD_LOGIC_1164.all;
```

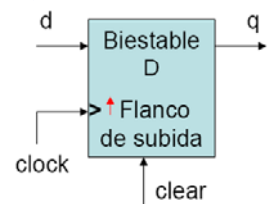
```
ENTITY biestable_D_con_clr IS
  PORT (clock,clear, d: IN STD_LOGIC; q: OUT STD_LOGIC);
END biestable_D_con_clr;
```

37

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXVIII)

2.3 FLIP FLOP D con entradas asíncronas (CLEAR) (XIV)

clock	clear	d(t)	q(t)	q(t+1)
inactivo	inactivo	x	x	q(t)
activo	inactivo	x	x	d(t)
x	activo	x	x	'0'



MODELADO CON VHDL

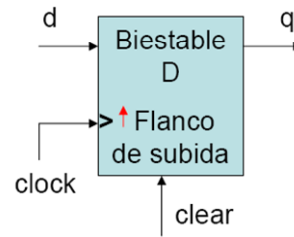
```
-- Arquitectura en estilo de COMPORTAMIENTO
-- Clear asíncrono y activo con valor bajo.
-- Se evalúa antes
-- Reloj Activo por flanco de subida [rising_edge (clock)]
```

```
-----
ARCHITECTURE comporta_asin OF biestable_D_con_clr IS
BEGIN
  PROCESS (clock, clear)
  BEGIN
    IF (clear='0') THEN q<= '0';
    ELSIF rising_edge (clock) THEN q<=d;
    END IF;
  END PROCESS;
END comporta_asin;
```

38

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXIX)

2.3 FLIP FLOP D con entradas asíncronas (CLEAR) (XV)



MODELADO CON VHDL

-- Arquitectura en estilo de FLUJO DE DATOS.

-- Clear asíncrono.

-- Reloj Activo por flanco de bajada

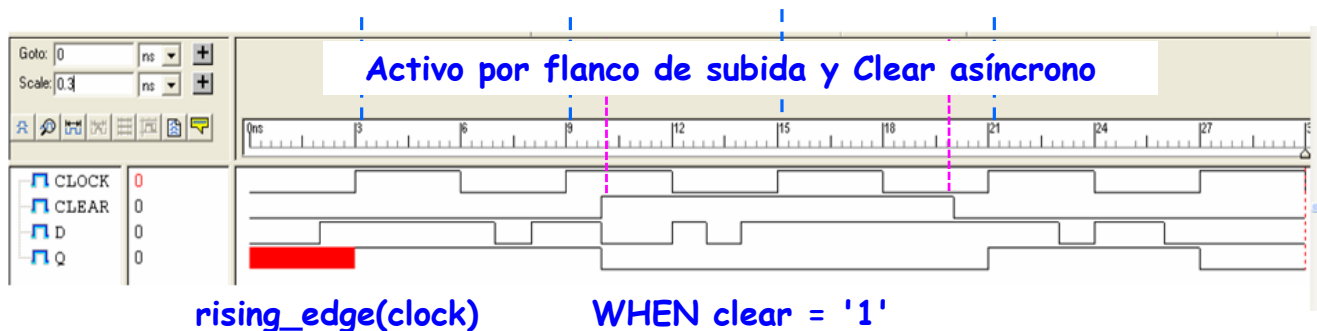
```
ARCHITECTURE flujo_asin OF biestable_D_con_clr IS
BEGIN
  q <= '0' WHEN clear = 'o' ELSE d WHEN falling_edge (clock);
END flujo_asin;
```

39

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXX)

2.3 FLIP FLOP D con entradas asíncronas (CLEAR) (XVI)

RESULTADOS DE SIMULACIÓN



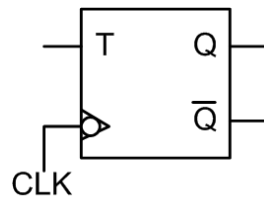
NOTA: Las líneas están ligeramente desplazadas para una visualización mejor

40

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXXI)

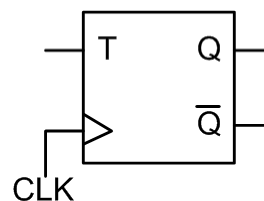
2.3 FLIP FLOP T (biestable T síncrono por flanco) (I)

FLANCO DE BAJADA ↘



CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↓	1	$\bar{Q}(t)$	Q(t)
↓	0	Q(t)	$\bar{Q}(t)$

FLANCO DE SUBIDA ↗



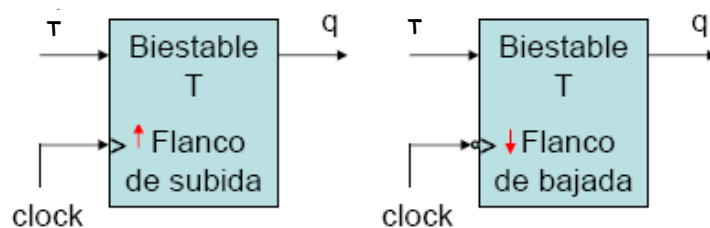
CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↑	1	$\bar{Q}(t)$	Q(t)
↑	0	Q(t)	$\bar{Q}(t)$

41

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXXII)

2.3 FLIP FLOP T (II)

MODELADO CON VHDL



CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↓	1	$\bar{Q}(t)$	Q(t)
↓	0	Q(t)	$\bar{Q}(t)$

CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↑	1	$\bar{Q}(t)$	Q(t)
↑	0	Q(t)	$\bar{Q}(t)$

--Entidad

LIBRARY IEEE;

USE ieee.STD_LOGIC_1164.all;

ENTITY biestable_T IS

GENERIC (retardo: TIME:= 0 ns);

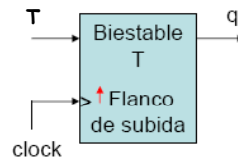
PORT (clock, t: IN STD_LOGIC; q: OUT STD_LOGIC);

END biestable_T;

42

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXXIII)

2.3 FLIP FLOP T (III)



CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↑	1	$\bar{Q}(t)$	Q(t)
↑	0	Q(t)	$\bar{Q}(t)$

MODELADO CON VHDL

-- Arquitectura en estilo de COMPORTAMIENTO

-- Reloj Activo por flanco de subida

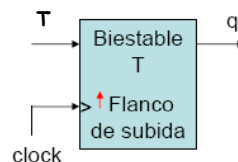
```

ARCHITECTURE comporta_biestable OF biestable_T IS
    SIGNAL valor_actual_q:STD_LOGIC:= '0';
BEGIN
    PROCESS (clock)
    BEGIN
        IF rising_edge (clock) THEN -- Flanco de subida
            IF (T=1) THEN valor_actual_q <= NOT valor_actual_q AFTER retardo
            ELSE NULL;
            END IF;
        END IF;
    END PROCESS;
END comporta_biestable;
    
```

43

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXIV)

2.3 FLIP FLOP T (IV)



CLK	T	Q(t+1)	$\bar{Q}(t+1)$
0	X	Q(t)	$\bar{Q}(t)$
1	X	Q(t)	$\bar{Q}(t)$
↑	1	$\bar{Q}(t)$	Q(t)
↑	0	Q(t)	$\bar{Q}(t)$

MODELADO CON VHDL

-- Arquitectura en estilo de FLUJO DE DATOS

-- Reloj Activo por flanco de subida

```

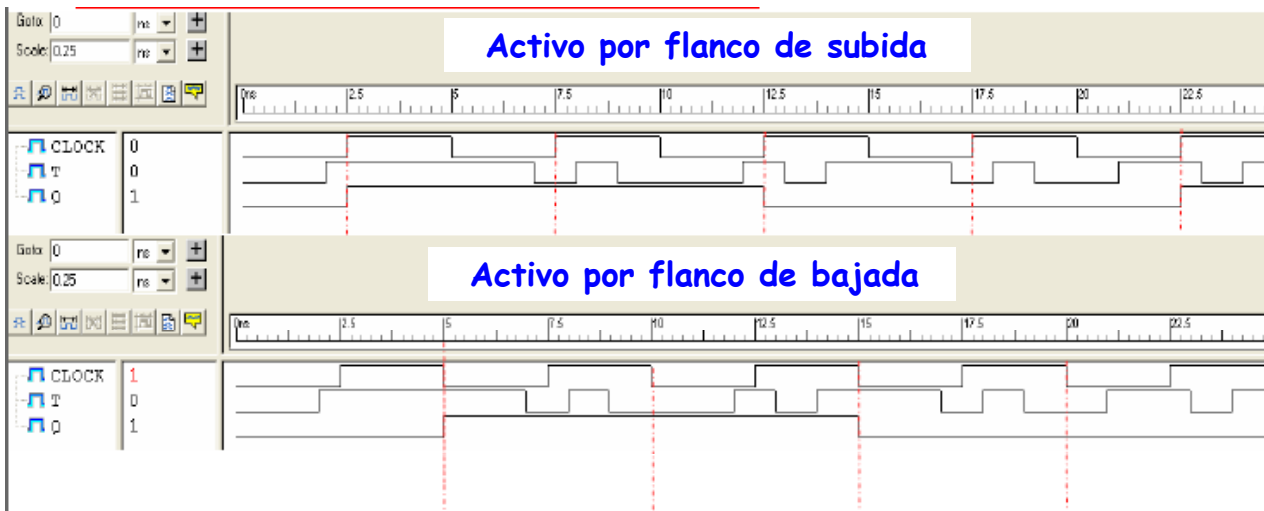
ARCHITECTURE flujo OF biestable_T IS
    SIGNAL valor_actual_q:STD_LOGIC:= '0';
BEGIN
    -- Activo por flanco de subida
    q<= valor_actual_q;
    Valor_actual_q <= NOT valor_actual_q AFTER retardo WHEN risig_edge
        (clock) AND T= '1' ELSE UNAFFECTED;
END flujo;
    
```

44

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXXV)

2.3 FLIP FLOP T (V)

RESULTADOS DE SIMULACIÓN



NOTA: si el valor de T es 1 durante la transición del reloj, la salida cambia; en caso contrario, permanece igual.

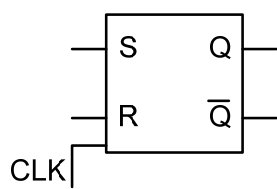
45

2. ALMACENAMIENTO ESTÁTICO DE LA INFORMACIÓN: LATCHES Y BIESTABLES (XXXVI)

CARACTERÍSTICAS GENERALES PARA EL MODELADO CON VHDL

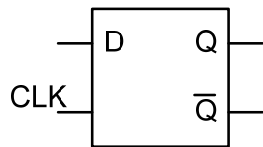
- Los elementos de memoria o almacenamiento se modelan con sentencias **PROCESS** (estilo de comportamiento) o con sentencias de asignación concurrente de señal (estilo de flujo de datos)
- Si se utilizan señales asíncronas (**CLEAR**), estas hay que incluirlas en la lista de sensibilidad del **PROCESS**

46

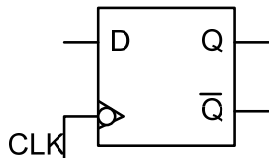


S	R	Q_{n+1}
0	0	Q_n
0	1	0
1	0	1
1	1	Ж

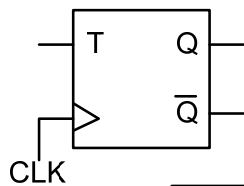
RESUMEN: tipos de biestables y su tabla de funcionamiento



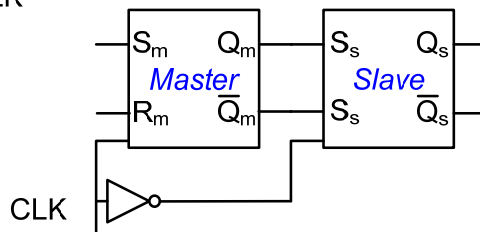
S	R	D	Q_{n+1}	Q_{n+1}
0	1	0	0	1
1	0	1	1	0



CLK	D	$Q(t+1)$	$\bar{Q}(t+1)$
0	X	$Q(t)$	$\bar{Q}(t)$
1	X	$Q(t)$	$\bar{Q}(t)$
↓	1	1	0
↓	0	0	1



CLK	T	$Q(t+1)$	$\bar{Q}(t+1)$
0	X	$Q(t)$	$\bar{Q}(t)$
1	X	$Q(t)$	$\bar{Q}(t)$
↑	1	$\bar{Q}(t)$	$Q(t)$
↑	0	$Q(t)$	$\bar{Q}(t)$



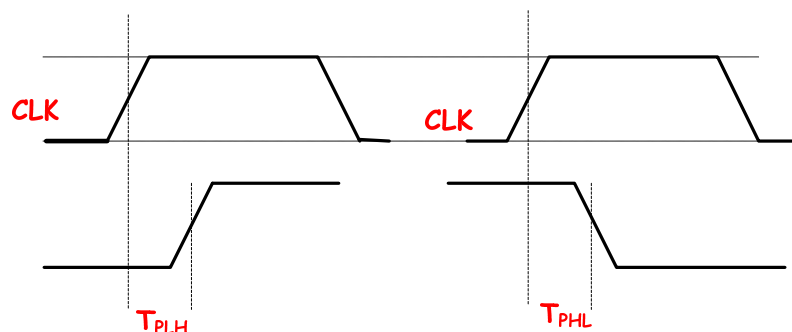
CLK	D	$Q(t+1)$	$\bar{Q}(t+1)$
0	X	$Q(t)$	$\bar{Q}(t)$
1	X	$Q(t)$	$\bar{Q}(t)$
↓	1	1	0
↓	0	0	1

47

3. ESPECIFICACIONES DE LOS BIESTABLES (I).

RETARDO DE PROPAGACIÓN T_R : intervalo de tiempo requerido para que se produzca un cambio en la salida, una vez que se ha aplicado una señal de entrada. Tiempo que tarda la salida de un circuito en responder a las modificaciones de la entrada

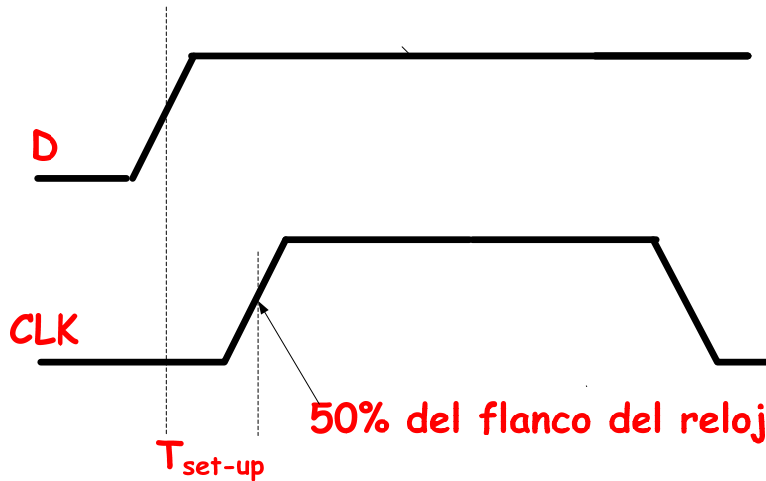
RETARDOS DE PROPAGACIÓN de baja a alta T_{PLH} y de alta a baja T_{PHL} : Tiempos en que tarda la salida de un circuito en pasar de baja a alta y de alta a baja respectivamente. Se mide desde el flanco del reloj hasta la transición, baja a alta, o alta a baja respectivamente.



48

3. ESPECIFICACIONES DE LOS BIESTABLES (II)

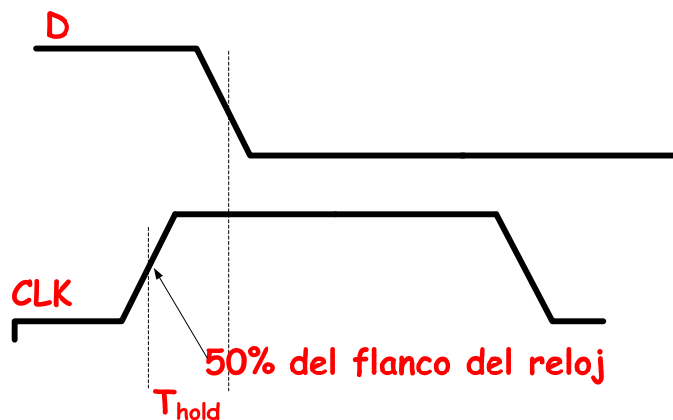
TIEMPO DE SETUP T_s : tiempo que precede a la activación del reloj durante el cual las señales de entrada tienen que mantenerse estables en el nivel adecuado



49

3. ESPECIFICACIONES DE LOS BIESTABLES (III)

TIEMPO DE HOLD T_H : tiempo que sigue a la transición activa del reloj durante el cual la señal de entrada tiene que mantenerse estable

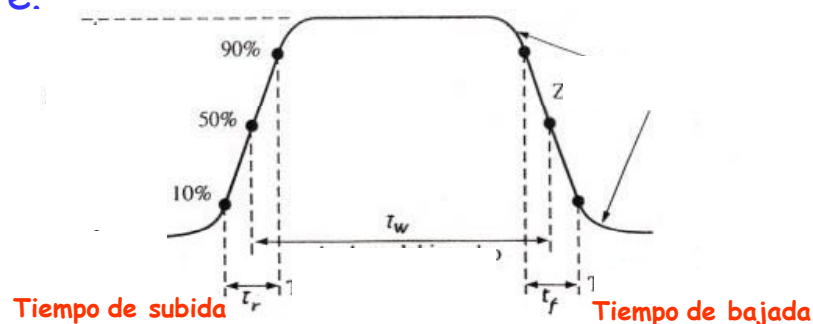


50

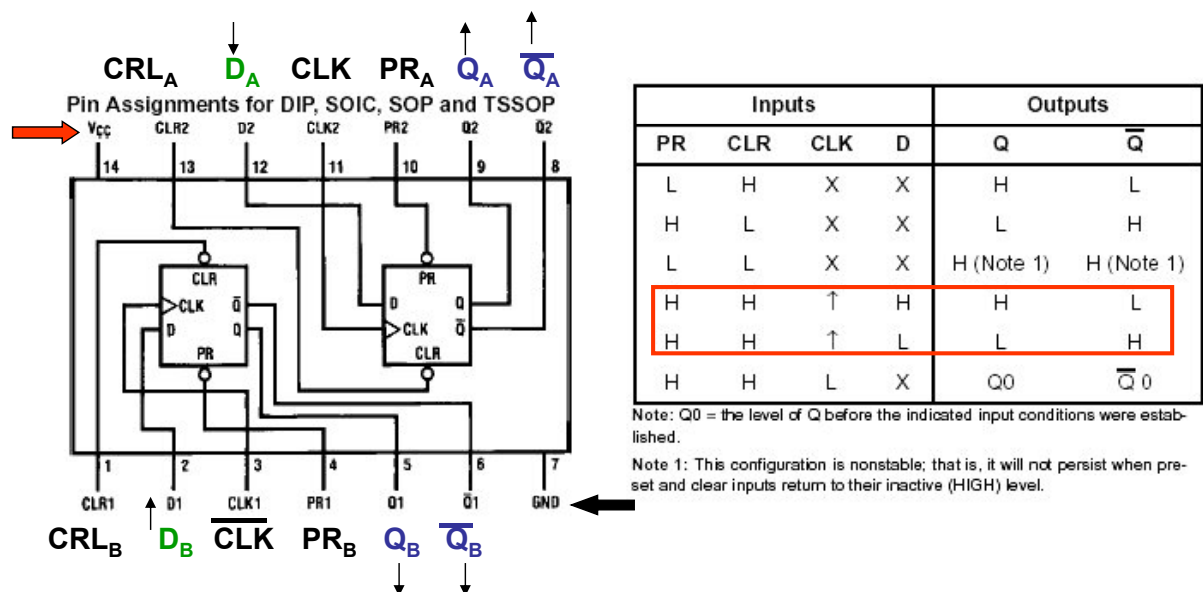
3. ESPECIFICACIONES DE LOS BIESTABLES (IV)

MAXIMA FRECUENCIA DE RELOJ F_{max} : es la máxima frecuencia que puede tener el reloj para garantizar un funcionamiento fiable del circuito. Es la mayor velocidad a la que se puede disparar el biestable de manera fiable

TIEMPOS DE SUBIDA T_R y CAÍDA (BAJADA) T_F : son los tiempos que tardan las señales aplicadas a los circuitos en pasar del 10 al 90% y del 90 al 10% de su valor total, respectivamente.



51



MM74HC74A

AC Electrical Characteristics					
$V_{CC} = 5V$, $T_A = 25^\circ C$, $C_L = 15 pF$, $t_r = t_f = 6 ns$					
Symbol	Parameter	Conditions	Typ	Guaranteed Limit	Units
f_{MAX}	Maximum Operating Frequency		72	30	MHz
t_{PHL}, t_{PLH}	Maximum Propagation Delay Clock to Q or \bar{Q}		10	30	ns
t_{PHL}, t_{PLH}	Maximum Propagation Delay Preset or Clear to Q or \bar{Q}		17	40	ns
t_{REM}	Minimum Removal Time, Preset or Clear to Clock		6	5	ns
t_s	Minimum Setup Time Data to Clock		10	20	ns
t_h	Minimum Hold Time Clock to Data		0	0	ns
t_W	Minimum Pulse Width Clock, Preset or Clear		8	16	ns

74HC74
2 Biestables D

52

4. REGISTROS DE DESPLAZAMIENTO UNI/BIDIRECCIONALES (I)

Registro: bloque funcional destinado a almacenar o registrar información binaria durante un cierto tiempo.

- Circuito temporal de memoria con capacidad limitada
- Consta de una cadena de *biestables* conectados en cascada de forma que la salida de uno es la entrada del siguiente (implementación lógica estática)

Desplazamiento de la información: derecha
izquierda
bidireccional

Entrada/salida de la información: serie/paralelo

53

4. REGISTROS DE DESPLAZAMIENTO UNI/BIDIRECCIONALES (II)

Tipos de registro atendiendo a la entrada/salida de la información:

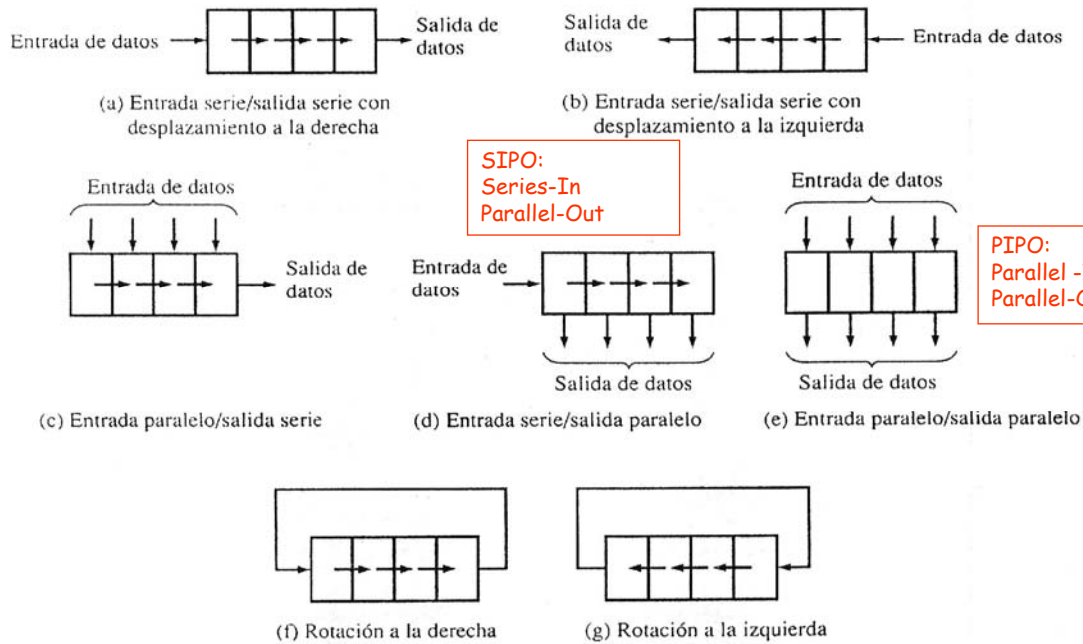
- ENTRADA SERIE/SALIDA SERIE
- ENTRADA SERIE/SALIDA PARALELO
- ENTRADA PARALELO/SALIDA SERIE
- ENTRADA PARALELO/SALIDA PARALELO

- BIDIRECCIONALES

54

4. REGISTROS DE DESPLAZAMIENTO UNI/BIDIRECCIONALES (III)

SISO:
Series-In Series-
Out

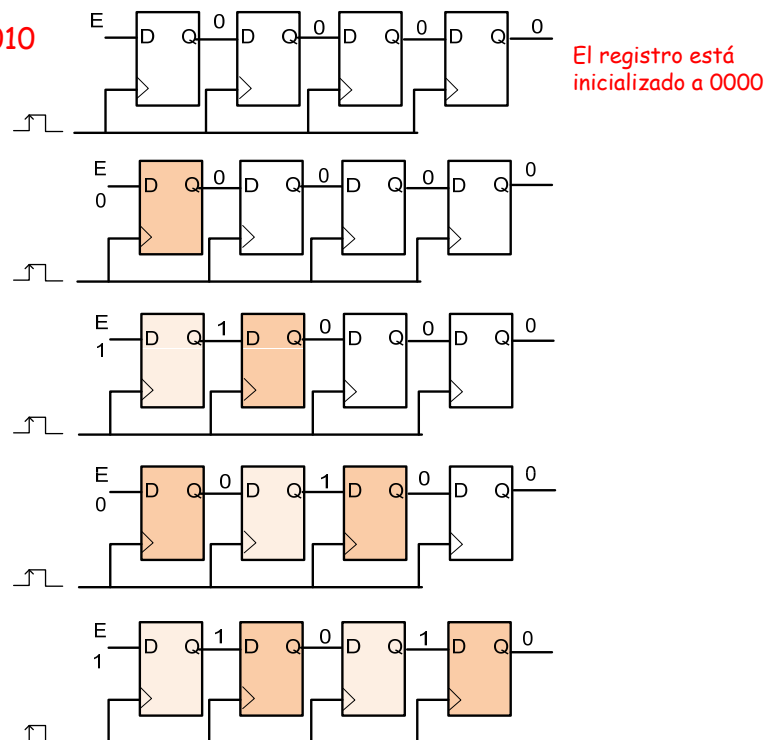


55

EJEMPLO de registro serie/serie, de 4 bits, síncrono, con biestables tipo D, disparados por flanco de subida.

Entrada:

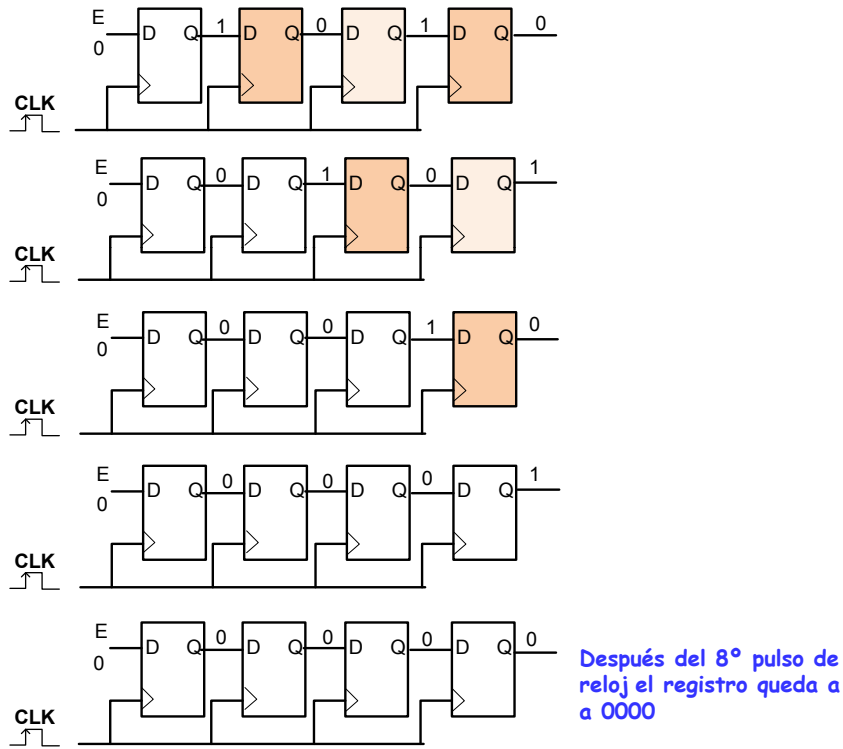
Secuencia de bits 1010



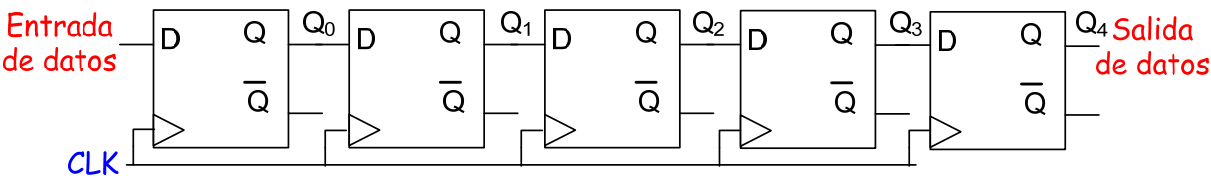
56

EJEMPLO de registro serie/serie, de 4 bits, síncrono, con biestables tipo D, disparados por flanco de subida.

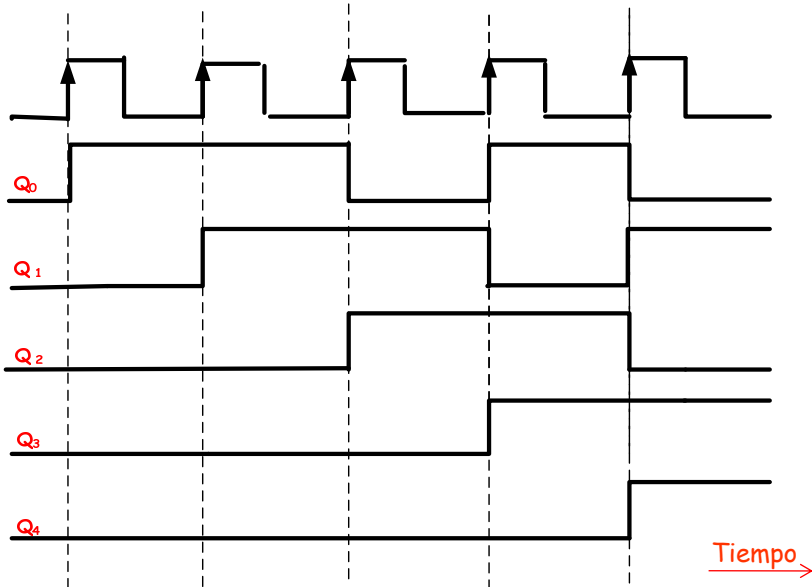
Salida:
Secuencia de bits 1010



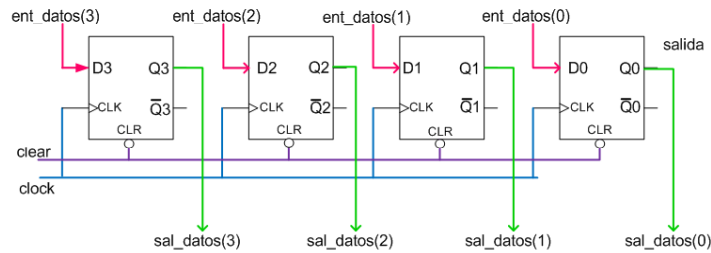
EJEMPLO: Obténgase el estado de un registro de desplazamiento de 5 bits para las señales de entrada de datos y señal de reloj representadas en la figura. Supóngase que inicialmente el contenido del registro es cero.



Datos de entrada:
11010 ←



EJEMPLO de registro paralelo/paralelo



EJEMPLO de registro paralelo/paralelo de 4 bits, síncrono, con biestables tipo D, disparados por flanco de subida.

```
-- Librerías y paquetes
-- Entidad
ENTITY NOMBRE_en IS
    GENERIC (num_bits:natural);
    PORT (CLK,CLR : IN STD_LOGIC;
          ent : -----;
          sal : -----;
    END NOMBRE_en;
-- Arquitectura
ARCHITECTURE NOMBRE_arq OF NOMBRE_en IS

    COMPONENT NOMBRE_componente IS
        GENERIC (retardo:TIME:=0ns);
        PORT Puertos_del_componente(BIESTABLE);
    END COMPONENT;
    BEGIN
        U1:FOR i IN 0 TO (num_bits - 1) GENERATE
            B1: NOMBRE_componente PORT MAP(CLK => , CLR => , D => ent(i), Q => sal(i));
        END GENERATE;
    END NOMBRE_arq;
```

En la librería deberíamos disponer de :
Biestable tipo D ↑ con señal CLEAR asíncrono

59

EJEMPLO de registro paralelo/paralelo. Arquitectura estructural

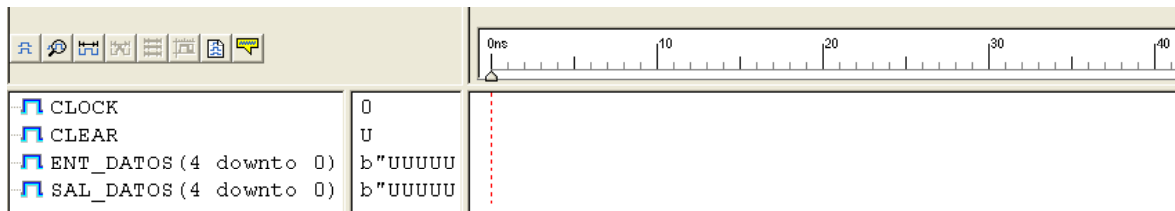
```
-----
-- © Universidad Politécnica de Madrid
-- Se permite copia para fines de estudio
-----
-- Proyecto      : Práctica 2: Biestables y registros
-- Diseño        : Registro paralelo paralelo
-- Nombre del fichero : R_paralelo_paralelo_gene.vhd
-- Autor         : MPC
-- Fecha         : 10/03/2014
-- Resumen       : Este fichero contiene la entidad y
--                 arquitectura de un registro paralelo paralelo construido con biestables tipo D
--                 activados por flanco de subida de reloj
--                 y clear asíncrono activo a nivel alto, utilizando la sentencia generate.
--                 El tipo de datos utilizado es STD_LOGIC
-----
-- Librerías y paquetes
LIBRARY IEEE;
USE ieee.STD_LOGIC_1164.all;
ENTITY registro_PP IS
    GENERIC (num_bits:natural);
    PORT (CLK,CLR : IN STD_LOGIC;
          ent : IN STD_LOGIC_VECTOR ((num_bits - 1) DOWNTO 0);
          sal : OUT STD_LOGIC_VECTOR ((num_bits - 1) DOWNTO 0));
END registro_PP;

ARCHITECTURE con_generate OF registro_PP IS
    COMPONENT biestable_d_con_clr IS
        GENERIC (retardo:TIME:=0ns);
        PORT (CLK,CLR, D: IN STD_LOGIC;
              Q : OUT STD_LOGIC);
    END COMPONENT;
    BEGIN
        U1:FOR i IN 0 TO (num_bits - 1) GENERATE
            B1: biestable_d_con_clr PORT MAP(CLK => CLK, CLR => CLR, D => ent(i), Q => sal(i));
        END GENERATE;
    END con_generate;
```

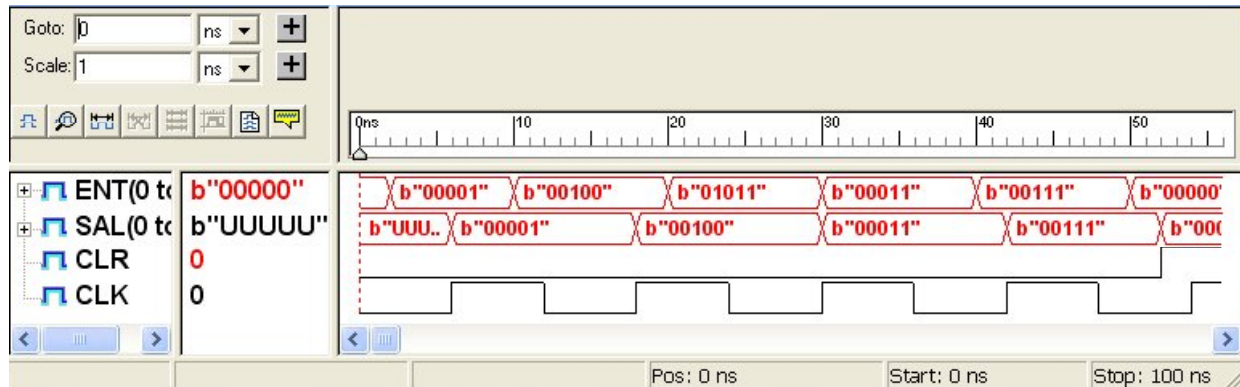
60

EJEMPLO de registro paralelo/paralelo. Simulación

Captura de pantalla del registro con las señales ubicadas en el orden siguiente: entradas_datos, salidas_datos, señal CLR y señal de reloj

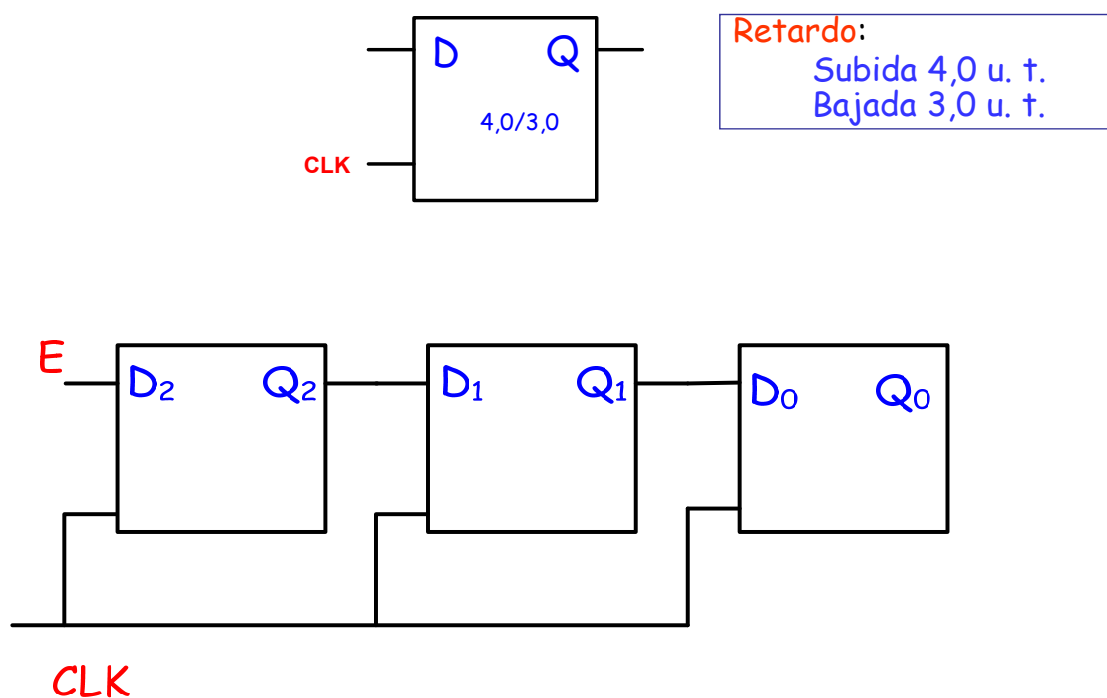


Captura de pantalla del registro con los resultados de una simulación

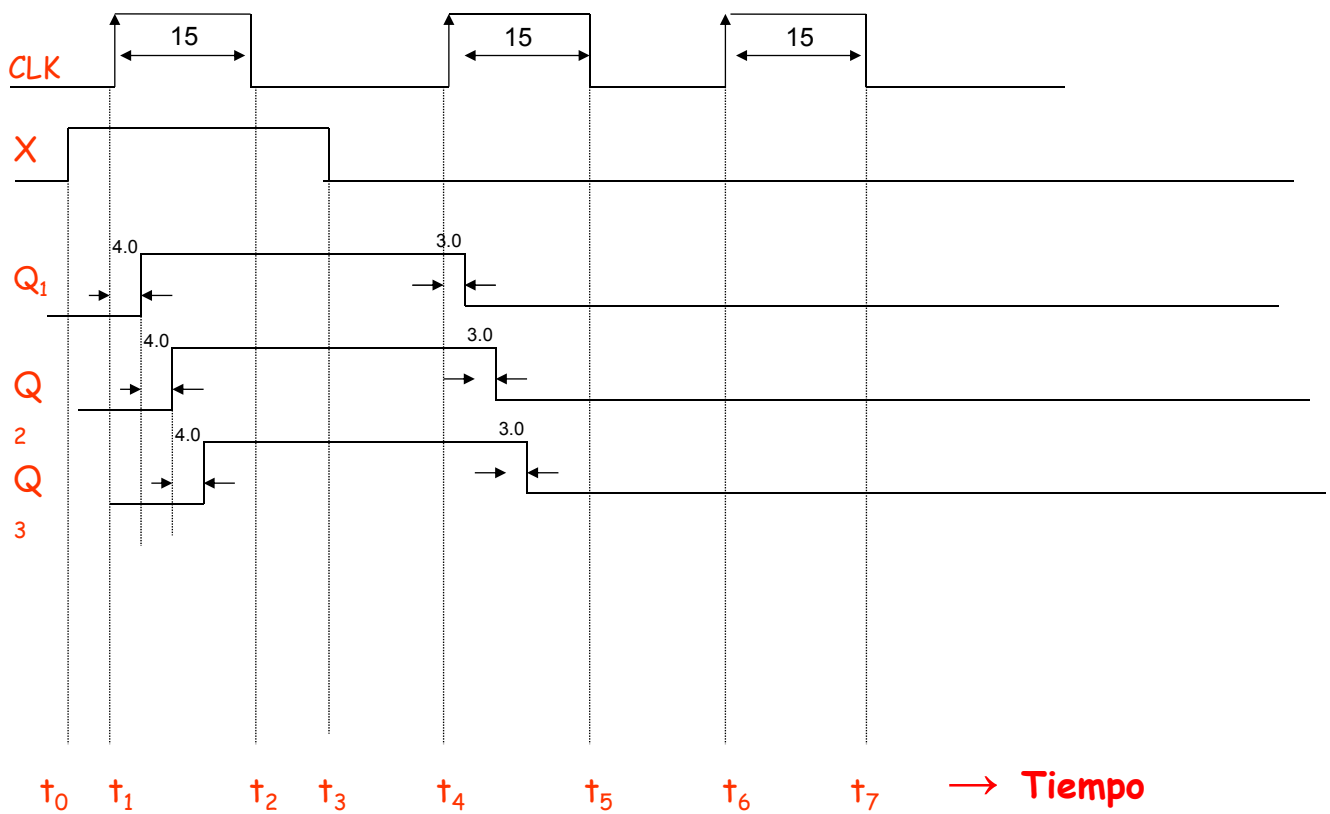


61

Ejemplo de "mal" uso de reloj por nivel: (resultado erróneo)

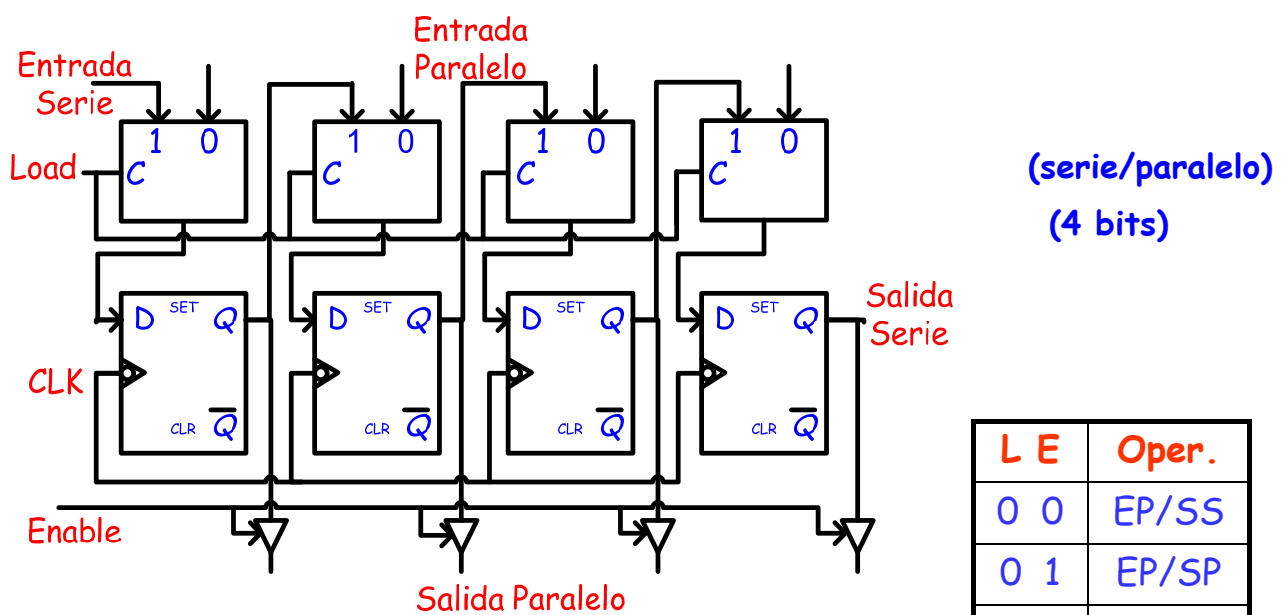


62



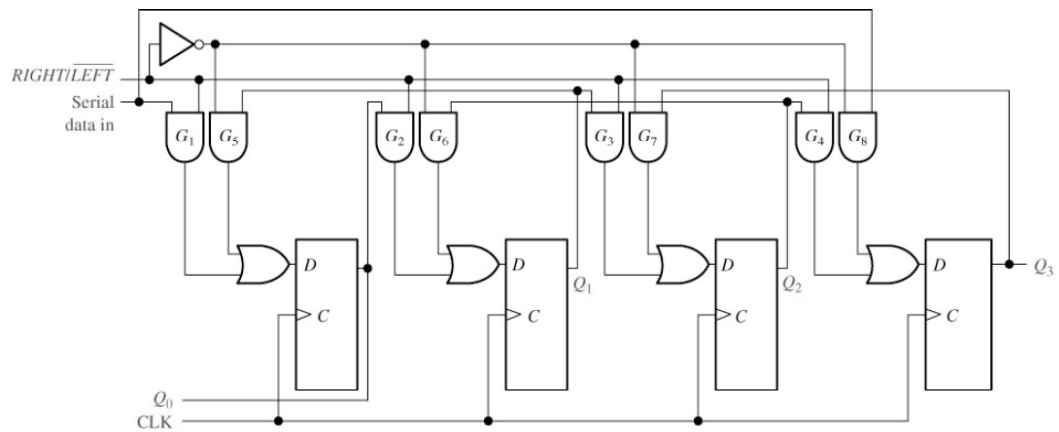
63

REGISTRO DE DESPLAZAMIENTO UNIVERSAL

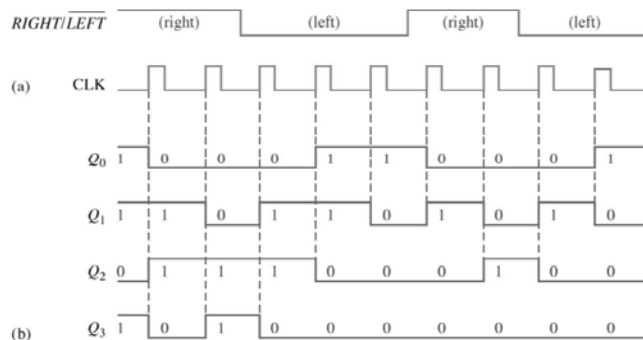


64

REGISTRO DE DESPLAZAMIENTO UNIVERSAL



Funcionamiento



65

5. PILAS FIFO Y LIFO (I)

Las pilas son memorias de acceso secuencial y se suelen construir utilizando n registros de desplazamiento de longitud m ; es decir están constituidas por biestables que permiten la carga y descarga de información.

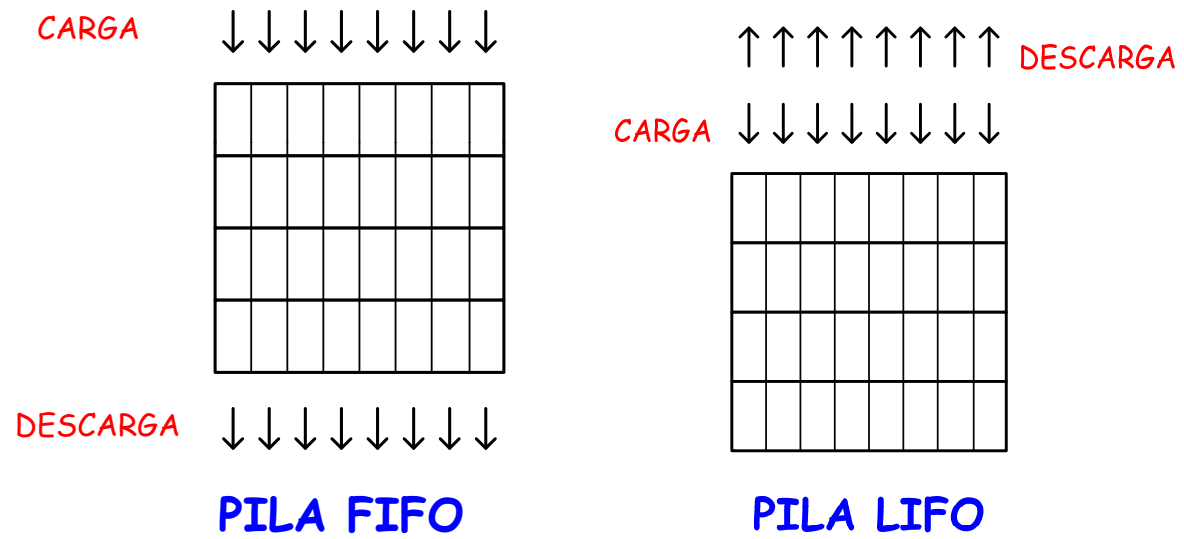
FIFO: First Input First Output

Es una memoria de acceso secuencial con dos puntos de acceso, uno para la lectura y otro para la escritura. Un apila FIFO, se construye con n registros de desplazamiento de longitud m , con desplazamientos hacia la derecha, mas la lógica de control necesaria.

LIFO: Last Input First Output

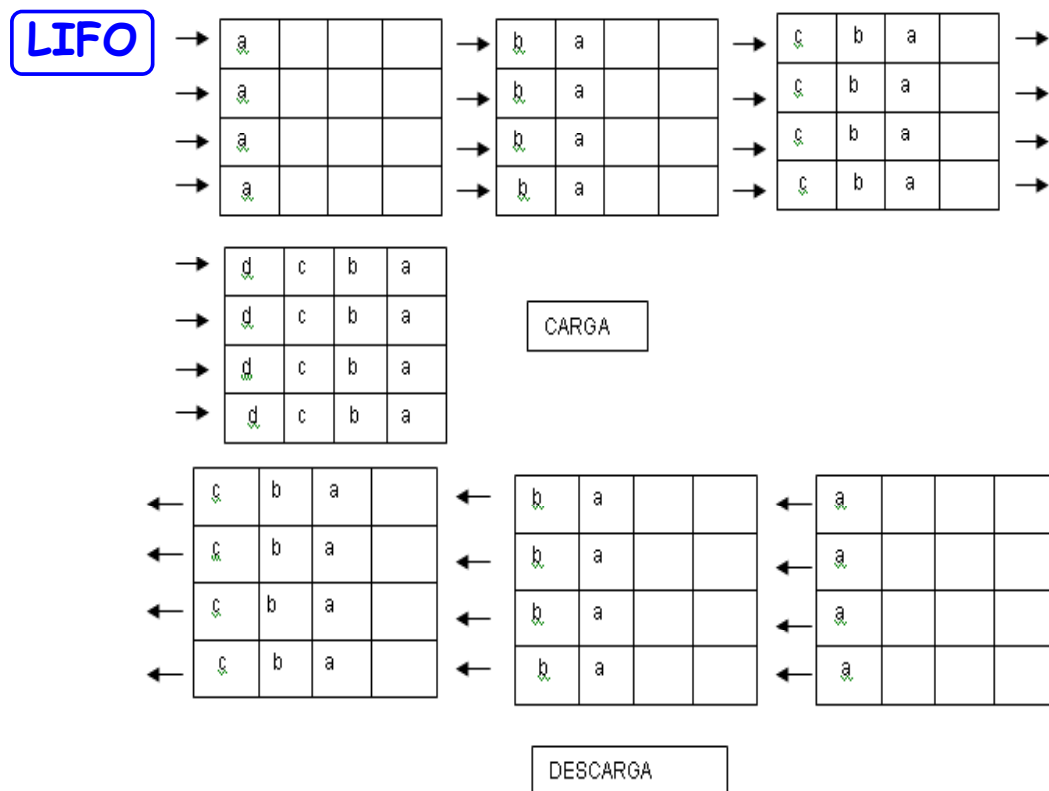
Es una memoria de acceso secuencial con un solo punto de acceso tanto para la lectura como para la escritura. Una pila LIFO que almacene m palabras de n bits, se construye utilizando n registros de desplazamiento de longitud m , con desplazamientos hacia la derecha y hacia la izquierda, mas lo lógica de control necesaria.

5. PILAS FIFO Y LIFO (II)



67

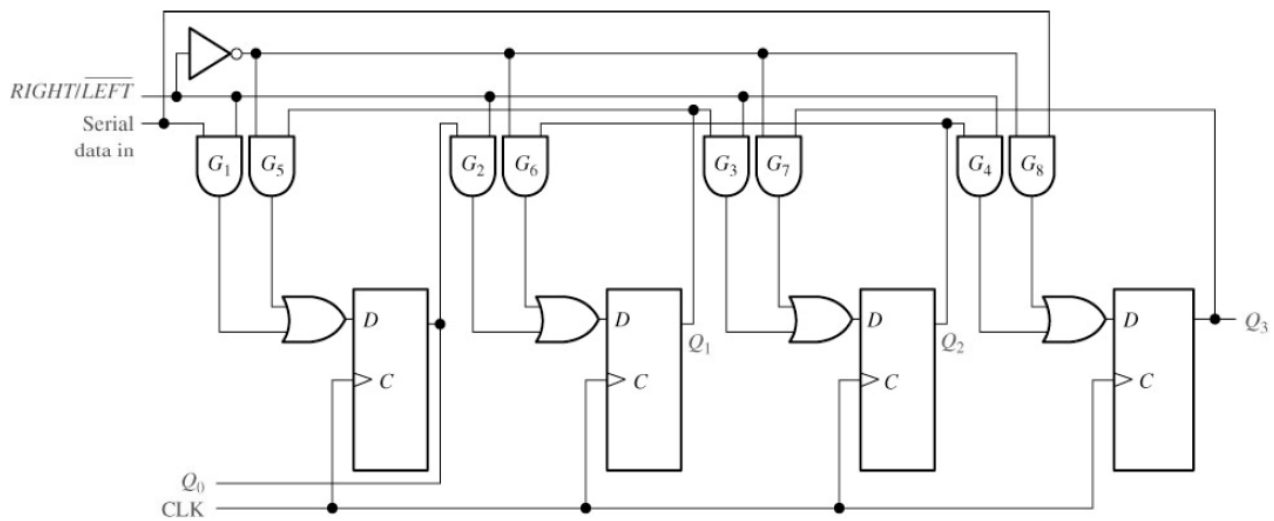
5. PILAS FIFO Y LIFO (III)



68

5. PILAS FIFO Y LIFO (IV)

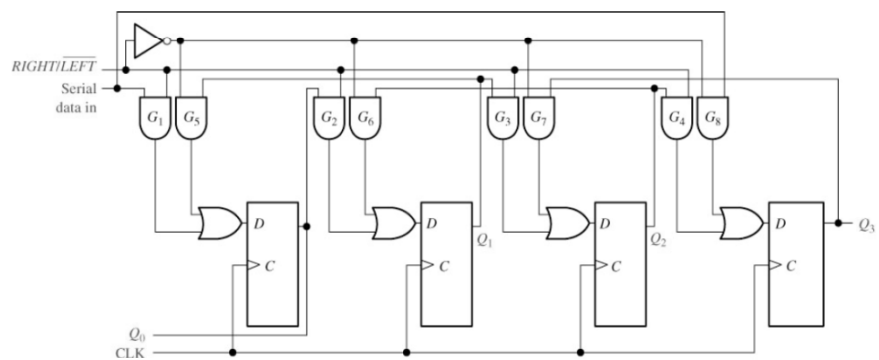
FIFO y LIFO



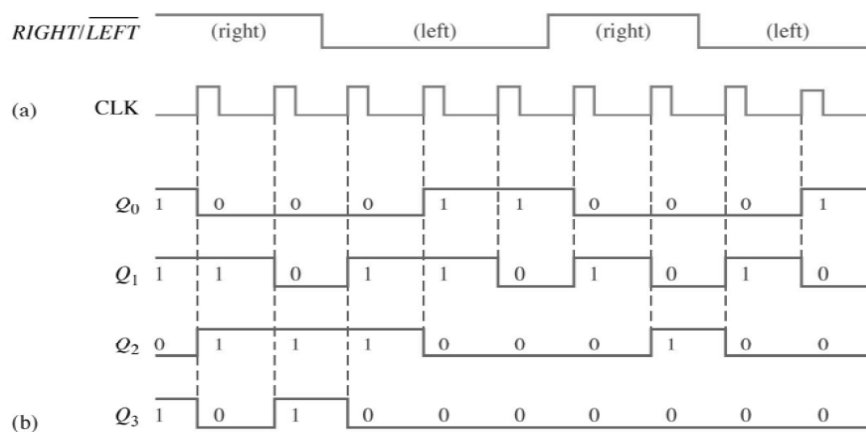
$\overline{RIGHT/LEFT} \rightarrow$ bidireccional

69

5. PILAS FIFO Y LIFO (V)



Funcionamiento



Esta pila es igual a un registro de desplazamiento bidireccional cuando solamente se compone de un bit (ver pg. 65)

70

6. CONTADORES (I)

- Los biestables pueden también conectarse entre si para hacer operaciones "de contar"
- El número de biestables que se utilizan y la forma en que se conectan determina el número de estados (**módulo**) y la secuencia específica por la que atraviesa el contador durante un ciclo completo.
- **EJEMPLO:** Un contador módulo 3 (hasta 7):
tiene 8 estados
para codificar 8 estados necesitamos
 $2^n \geq 8 \rightarrow n = 3$ biestables

71

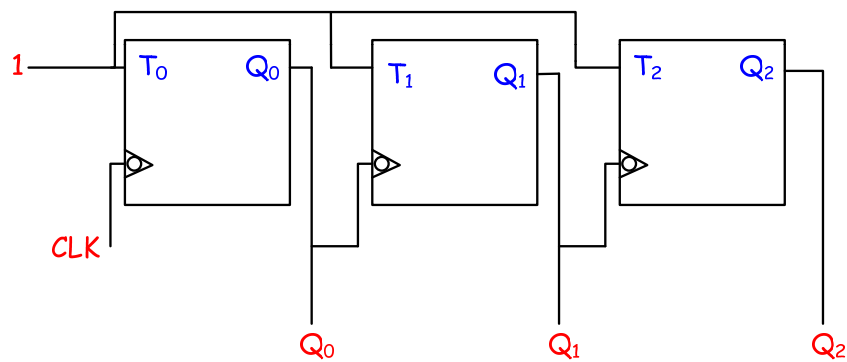
6. CONTADORES (II)

- Dependiendo del modo de la señal de reloj, los contadores se clasifican en : ASÍNCRONOS Y SÍNCRONOS.
 - ASÍNCRONOS: en estos se aplica una señal de reloj en la entrada de reloj del primer biestable y al resto de los biestables se conecta la salida Q de la etapa anterior. Los biestables no cambian de estado simultáneamente
 - SÍNCRONOS: la señal de reloj se aplica a todos los biestables; trabajan simultáneamente

72

6. CONTADORES (III)

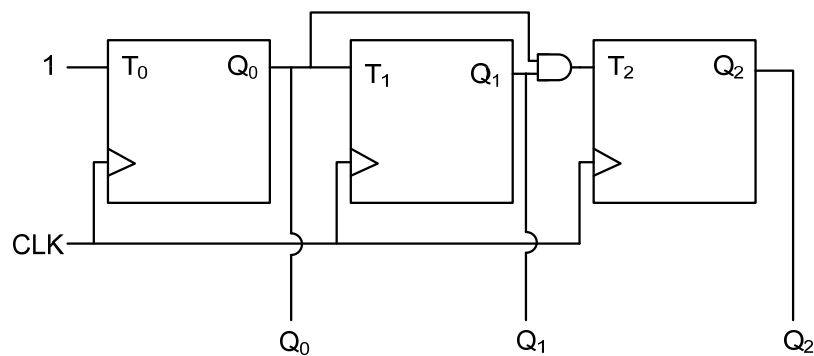
CONTADOR ASÍNCRONO



Valor actual			Valor siguiente		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

6. CONTADORES (IV)

CONTADOR SÍNCRONO

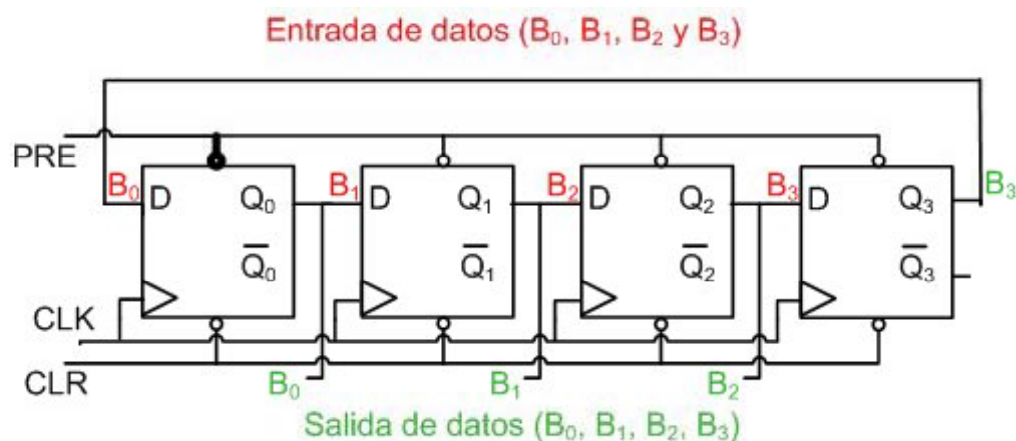


Valor actual			Valor siguiente		
Q ₂	Q ₁	Q ₀	Q ₂	Q ₁	Q ₀
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0
0	0	0	0	0	1
0	0	1	0	1	0
0	1	0	0	1	1
0	1	1	1	0	0
1	0	0	1	0	1
1	0	1	1	1	0
1	1	0	1	1	1
1	1	1	0	0	0

CONTADORES EN ANILLO (V)

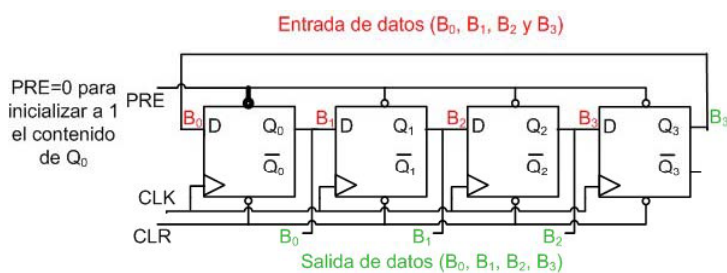
Un contador en anillo, cíclico y síncrono se construye concatenando biestables de forma similar a la construcción de un registro de desplazamiento, pero en el caso de los contadores, la salida del último elemento de la cadena de biestables se conecta con la entrada del primero.

Contador cíclico
de 4 bits.
Modulo 4
4 estados



75

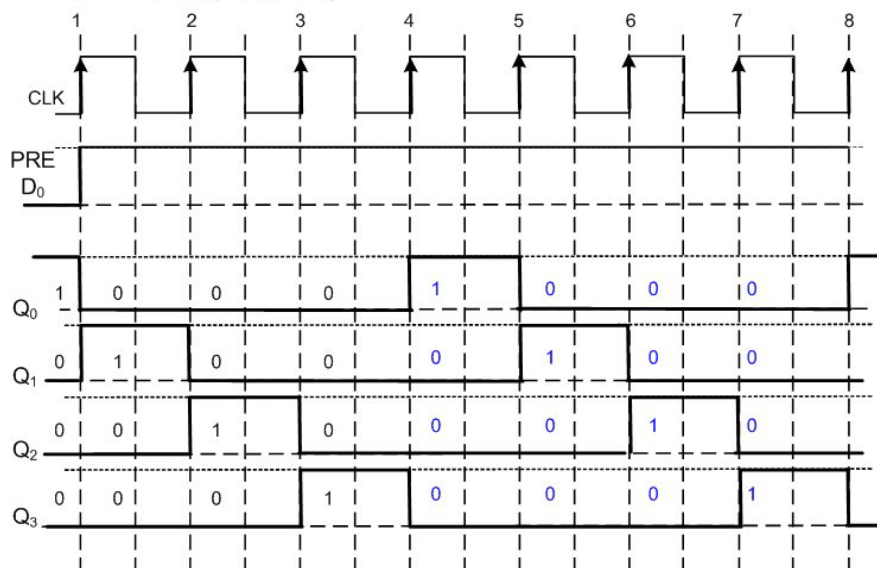
CONTADORES EN ANILLO (VI)



CONTENIDO

1000
0100
0010
0001
1000

OPERACIÓN QUE HACE:
Empieza en 8 y va
dividiendo por 2, cuando ya
no puede seguir, empieza
otra vez por 8....



CRONOGRAMA CON EL CONTENIDO DE LOS BIESTABLES

76

CONTADORES EN ANILLO: Contador Johnson (VII):

La salida **negada** del último elemento se conecta con la entrada

